Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2016-006 May 26, 2016

Towards Practical Theory: Bayesian Optimization and Optimal Exploration

Kenji Kawaguchi

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Towards Practical Theory: Bayesian Optimization and Optimal Exploration

by

Kenji Kawaguchi

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2016

Author .....
Department of Electrical Engineering and Computer Science
January 15, 2016

Certified by .....
Leslie P. Kaelbling
Professor of Computer Science and Engineering
Thesis Supervisor

Certified by .....
Tomas Lozano-Perez
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by .....
Professor Leslie A. Kolodziejski
Chair of the Committee on Graduate Students

Towards Practical Theory: Bayesian Optimization and Optimal Exploration

by

Kenji Kawaguchi

Submitted to the Department of Electrical Engineering and Computer Science
on January 15, 2016, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

This thesis discusses novel principles to improve the theoretical analyses of a class of methods, aiming to provide theoretically driven yet practically useful methods. The thesis focuses on a class of methods, called bound-based search, which includes several planning algorithms (e.g., the A* algorithm and the UCT algorithm), several optimization methods (e.g., Bayesian optimization and Lipschitz optimization), and some learning algorithms (e.g., PAC-MDP algorithms). For Bayesian optimization, this work solves an open problem and achieves an exponential convergence rate. For learning algorithms, this thesis proposes a new analysis framework, called PAC-RMDP, and improves the previous theoretical bounds. The PAC-RMDP framework also provides a unifying view of some previous near-Bayes optimal and PAC-MDP algorithms. All proposed algorithms derived on the basis of the new principles produced competitive results in our numerical experiments with standard benchmark tests.

Thesis Supervisor: Leslie P. Kaelbling
Title: Professor of Computer Science and Engineering

Thesis Supervisor: Tomas Lozano-Perez
Title: Professor of Computer Science and Engineering

Acknowledgments

I would like to express my sincere gratitude to my advisors Prof. Leslie Kaelbling and Prof. Tomás Lozano-Pérez, for all of their support and keen insights. In addition to the technical input provided by them, my experience of their quick, flexible, and profound thinking and discussions has been valuable, providing a sense of successful research styles that I could learn from.

Further, I would like to thank Dr. Remi Munos for his thoughtful comments and suggestions for my work on Bayesian optimization. Without his previous work on optimistic optimization, this work could not have been completed.

I would also like to thank Prof. Michael Littman for his thoughtful comments on and suggestions for my work on optimal exploration in MDP. His suggestions and encouraging comments motivated me to complete the work.

Next, I would like to acknowledge Funai Overseas Scholarship for their very generous financial support and for providing me with a scientific community active in universities across the world.

Finally, I would thank my wife Molly Kruko for her support and happy moments that we share, which have been a vital source of motivation and relaxation to me while completing this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"In theory, there is no difference between theory and practice. But, in practice, there is."– Jan L. A. van de Snepscheut. One of the reasons why theory and practice can diverge is that a set of assumptions made in theory can be invalid in practice. The well-known *spherical cow* metaphor exemplifies this phenomenon (i.e., theoretical results based on the assumptions of the spherical shape and the vacuum condition may not be directly useful in practice for farmers). Indeed, it is well recognized that we should carefully select a set of *valid* assumptions that hold in practice. However, selecting a set of *valid* assumptions is often not sufficient to prevent the divergence of theory and practice. The divergence can occur when a set of assumptions are not sufficiently *sharp* (or tight) to exclude irrelevant problems or phenomena while capturing the class of the problems at hand.

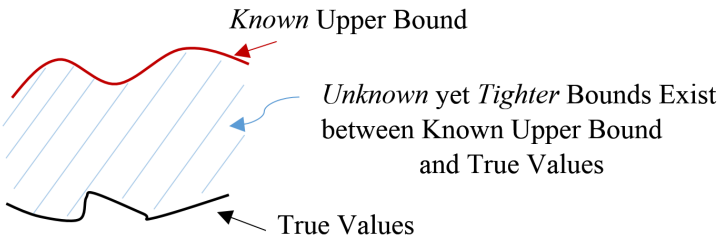
In this thesis, I first identify a class of methods that has suffered from this *sharpness* issue at an algorithmic level, resulting in the divergence of theory and practice. I refer to the identified method class as *bound-based search*, which includes the A* search, Upper Confidence for Trees (UCT), and Forward Search Sparse Sampling (FSSS) algorithms [50]; exploration in Markov decision processes (MDPs) with optimism in the face of uncertainty [19]; Lipschitz optimization [37, 24, 7]; and Bayesian optimization with an upper confidence bound [41, 10]. Bound-based search methods have a common property: The tightness of the bound determines its effectiveness. The tighter the bound is, the better is the performance. However, it is often difficult

to obtain a tight bound while maintaining correctness in a theoretically supported manner. For example, in A* search, admissible heuristics maintain the correctness of the bound, but the estimated bound with admissibility is often very loose in practice, resulting in long execution times for global search. This has seemingly lead to the divergence of theoretically driven approaches (that focus more on global search, maintaining theoretical guarantees yet taking a long time to converge in practice) and practically driven ones (that focus more on local search, obtaining a reasonable solution within a short time but with no global theoretical guarantee).

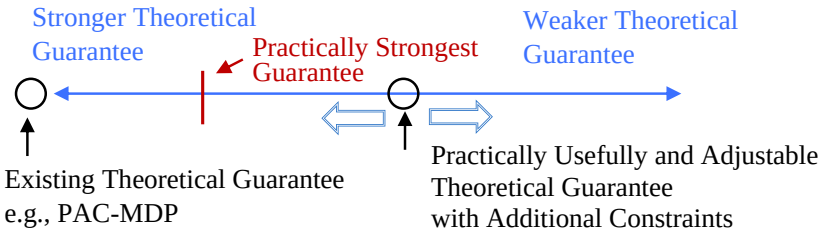
In this study, I have focused on two members of the class of bound-based search methods—Bayesian optimization and PAC–MDP algorithms—and proposed a specific solution for each one to tighten their bounds with sharper assumptions, aiming to provide theoretically driven algorithms that perform well in practice.

Our two high-level approaches are illustrated in Figure 1-1. For the approach shown in Figure 1-1 (a), note that *tighter* yet *unknown* bounds exist unless the known bound is *exact*. We propose a way to additionally take advantage of this simple fact that has been overlooked in previous work. A high-level idea is the following. When we rely on a single known bound, the set of the next evaluation candidates is totally ordered in terms of the possible improvements with respect to the bound. In contrast, we consider a set of unknown tighter bounds (possibly uncountable), and as a result, the set of the next evaluation candidates can be partially ordered but not totally ordered in terms of the improvements with respect to the set of the bounds. Thus, we simultaneously evaluates several minimal (or maximal) elements of the partially ordered set to account for the existence of the unknown yet tighter bounds.

Another approach, illustrated in Figure 1-1 (b), is based on two ideas. First, there exist practical constraints that the current theory ignores. Because of that, what the current theory guarantees (the leftmost circle in Figure 1-1 (b)) can be considerably stronger than necessary (the red line in Figure 1-1 (b)). This has been a partial cause of loose theoretical bounds and impractical algorithms. Thus, we reduce the size of the practically irrelevant region in the scope of theoretical analyses. Second, we propose a way to adjust the degree of theoretical guarantees in order to achieve



(a) An approach for Bayesian Optimization



(b) An approach for Exploration in MDP

Figure 1-1: Two distinct approaches to improve the theory for bound-based search methods: For Bayesian optimization, I have leveraged existence of unknown yet tighter bounds. For the MDP exploration problem, I have proposed an adjustable theoretical guarantee to accommodate practical needs.

the practicality that we want (the right circle in Figure 1-1 (b)).

1.1 Bayesian Optimization

I consider a general constrained global optimization problem: maximize $f(x)$ subject to $x \in \Omega \subset \mathbb{R}^D$, where $f : \Omega \to \mathbb{R}$ denotes a non-convex black-box deterministic function. Such a problem arises in many real-world applications, such as parameter tuning in machine learning [39], engineering design problems [8], and model parameter fitting in biology [55]. For this problem, one performance measure of an algorithm is the *simple regret*, r_n, which is given by $r_n = \sup_{x\in\Omega} f(x) - f(x^+)$ where x^+ is the best input vector found by the algorithm. For brevity, I use the term "regret" to mean simple regret.

The general global optimization problem is known to be intractable if we make no further assumptions [11]. The simplest additional assumption to restore tractability

is to assume the existence of a bound on the slope of f. A well-known variant of this assumption is Lipschitz continuity with a known Lipschitz constant, and many algorithms have been proposed in this setting [37, 28, 30]. These algorithms successfully guarantee certain bounds on the regret. However appealing from a theoretical point of view, a practical concern was soon raised regarding the assumption that a tight Lipschitz constant is known. Some researchers relaxed this somewhat strong assumption by proposing procedures for estimating a Lipschitz constant during the optimization process [47, 24, 7].

Bayesian optimization is an efficient way to relax this assumption of the complete knowledge of the Lipschitz constant, and has become a well-recognized method for solving global optimization problems with non-convex black-box functions. In the machine learning community, Bayesian optimization, particularly by means of a Gaussian process (GP), is an active research area [14, 52, 41]. With the requirement of the access to the δ-cover sampling procedure (which samples the function uniformly such that the density of the samples doubles in the feasible regions at each iteration), De Freitas et al. [10] recently proposed a theoretical procedure that maintains an exponential convergence rate (exponential regret). However, as pointed out by Wang et al. [51], the δ-cover sampling procedure is somewhat impractical. Accordingly, their paper states that one remaining problem is to derive a GP-based optimization method with an exponential convergence rate *without* the δ-cover sampling procedure.

In this thesis, I propose a novel GP-based global optimization algorithm, which maintains an exponential convergence rate and converges rapidly *without* the δ-cover sampling procedure. These results are described in Chapter 2 and a published paper [20].

1.2 Learning/Exploration in MDPs

The formulation of sequential decision making as an MDP has been successfully applied to a number of real-world problems. MDPs provide the ability to design adaptable agents that can operate effectively in uncertain environments. In many

situations, the environment that we wish to model has unknown aspects, and therefore, the agent needs to learn an MDP by interacting with the environment. In other words, the agent has to *explore* the unknown aspects of the environment to learn the MDP. A considerable amount of theoretical work on MDPs has focused on efficient exploration, and a number of principled methods have been derived with the aim of learning an MDP to obtain a near-optimal policy. For example, Kearns and Singh [22] and Strehl and Littman [43] considered discrete state spaces, whereas Bernstein and Shimkin [5] and Pazis and Parr [33] examined continuous state spaces.

In practice, however, heuristics are still commonly used [26]. This is partly because theoretically-driven methods tend to result in over-exploration. The focus of theoretical work (learning a near-optimal policy within a polynomial but long period of time) has apparently diverged from practical needs (learning a satisfactory policy within a reasonable period of time). In this thesis, I have modified the prevalent theoretical approach to develop theoretically-driven methods that come close to practical needs. These results are described in Chapter 2 and a published paper [18].

Chapter 2

Bayesian Optimization with Exponential Convergence

This chapter presents a Bayesian optimization method with exponential convergence *without* the need of auxiliary optimization and *without* the δ-cover sampling. Most Bayesian optimization methods require auxiliary optimization: an additional non-convex global optimization problem, which can be time-consuming and hard to implement in practice. Also, the existing Bayesian optimization method with exponential convergence [10] requires access to the δ-cover sampling, which was considered to be impractical [10, 51]. Our approach eliminates both requirements and achieves an exponential convergence rate.

2.1 Gaussian Process Optimization

In Gaussian process optimization, we estimate the distribution over function f and use this information to decide which point of f should be evaluated next. In a parametric approach, we consider a parameterized function $f(x;\theta)$, with θ being distributed according to some prior. In contrast, the nonparametric GP approach directly puts the GP prior over f as $f(\cdot) \sim GP(m(\cdot), \kappa(\cdot,\cdot))$ where $m(\cdot)$ is the mean function and $\kappa(\cdot,\cdot)$ is the covariance function or the kernel. That is, $m(x) = \mathbb{E}[f(x)]$ and $\kappa(x,x') = \mathbb{E}[(f(x)-m(x))(f(x')-m(x'))^T]$. For a finite set of points, the GP

model is simply a joint Gaussian: $\mathbf{f}(\mathbf{x}_{1:N}) \sim \mathcal{N}(\mathbf{m}(\mathbf{x}_{1:N}), \mathbf{K})$, where $\mathbf{K}_{i,j} = \kappa(x_i, x_j)$ and N is the number of data points. To predict the value of f at a new data point, we first consider the joint distribution over f of the old data points and the new data point:

$$\begin{pmatrix} \mathbf{f}(\mathbf{x}_{1:N}) \\ f(x_{N+1}) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m}(\mathbf{x}_{1:N}) \\ m(x_{N+1}) \end{pmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & \kappa(x_{N+1}, x_{N+1}) \end{bmatrix} \right)$$

where $\mathbf{k} = \kappa(\mathbf{x}_{1:N}, \mathbf{x}_{N+1}) \in \mathbb{R}^{N \times 1}$. Then, after factorizing the joint distribution using the Schur complement for the joint Gaussian, we obtain the conditional distribution, conditioned on observed entities $\mathcal{D}_N := \{\mathbf{x}_{1:N}, \mathbf{f}(\mathbf{x}_{1:N})\}$ and x_{N+1}, as:

$$f(\mathbf{x}_{N+1}) | \mathcal{D}_N, x_{N+1} \sim \mathcal{N}(\mu(x_{N+1}|\mathcal{D}_N), \sigma^2(x_{N+1}|\mathcal{D}_N))$$

where $\mu(x_{N+1}|\mathcal{D}_N) = m(x_{N+1}) + \mathbf{k}^T\mathbf{K}^{-1}(\mathbf{f}(\mathbf{x}_{1:N}) - \mathbf{m}(\mathbf{x}_{1:N}))$ and $\sigma^2(x_{N+1}|\mathcal{D}_N) = \kappa(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}$. One advantage of GP is that this closed-form solution simplifies both its analysis and implementation.

To use a GP, we must specify the mean function and the covariance function. The mean function is usually set to be zero. With this zero mean function, the conditional mean $\mu(x_{N+1}|\mathcal{D}_N)$ can still be flexibly specified by the covariance function, as shown in the above equation for μ. For the covariance function, there are several common choices, including the Matern kernel and the Gaussian kernel. For example, the Gaussian kernel is defined as $\kappa(x, x') = \exp\left(-\frac{1}{2}(x - x')^T\Sigma^{-1}(x - x')\right)$ where Σ^{-1} is the kernel parameter matrix. The kernel parameters or hyperparameters can be estimated by empirical Bayesian methods [32]; see [35] for more information about GP.

The flexibility and simplicity of the GP prior make it a common choice for continuous objective functions in the Bayesian optimization literature. Bayesian optimization with GP selects the next query point that optimizes the acquisition function generated by GP. Commonly used acquisition functions include the upper confidence bound (UCB) and expected improvement (EI). For brevity, we consider Bayesian op-

timization with UCB, which works as follows. At each iteration, the UCB function $\mathcal{U}$ is maintained as $\mathcal{U}(x|D_N) = \mu(x|\mathcal{D}_N) + \varsigma\sigma(x|\mathcal{D}_N)$ where $\varsigma \in \mathbb{R}$ is a parameter of the algorithm. To find the next query x_{n+1} for the objective function f, GP-UCB solves an additional non-convex optimization problem with $\mathcal{U}$ as $x_{N+1} = \arg\max_x \mathcal{U}(x|D_N)$. This is often carried out by other global optimization methods such as DIRECT and CMA-ES. The justification for introducing a new optimization problem lies in the assumption that the cost of evaluating the objective function f dominates that of solving additional optimization problem.

For deterministic function, de Freitas et al. [10] recently presented a theoretical procedure that maintains exponential convergence rate. However, their own paper and the follow-up research [10, 51] point out that this result relies on an impractical sampling procedure, the δ-cover sampling. To overcome this issue, Wang et al. [51] combined GP-UCB with a hierarchical partitioning optimization method, the SOO algorithm [31], providing a regret bound with polynomial dependence on the number of function evaluations. They concluded that creating a GP-based algorithm with an *exponential* convergence rate *without* the impractical sampling procedure remained an open problem.

2.2 Infinite-Metric GP Optimization

2.2.1 Overview

The GP-UCB algorithm can be seen as a member of the class of bound-based search methods, which includes Lipschitz optimization, A* search, and PAC-MDP algorithms with optimism in the face of uncertainty. As discussed in section 1, it is often difficult to obtain a tight bound for bound-based methods, resulting in a long period of global search or heuristic approach with no theoretical guarantee.

The GP-UCB algorithm has the same problem. The bound in GP-UCB is represented by UCB, which has the following property: $f(x) \leq \mathcal{U}(x|\mathcal{D})$ with some probability. I formalize this property in the analysis of our algorithm. The problem is essentially due to the difficulty of obtaining a tight bound $\mathcal{U}(x|\mathcal{D})$ such that

$f(x) \leq \mathcal{U}(x|\mathcal{D})$ and $f(x) \approx \mathcal{U}(x|\mathcal{D})$ (with some probability). Our solution strategy is to first admit that the bound encoded in GP prior may not be tight enough to be useful by itself. Instead of relying on a single bound given by the GP, I leverage the existence of an *unknown* bound encoded in the continuity at a global optimizer.

Assumption 2.1. (Unknown Bound) There exists a global optimizer x^* and an *unknown* semi-metric ℓ such that for all $x \in \Omega$, $f(x^*) \leq f(x) + \ell(x, x^*)$ and $\ell(x, x^*) < \infty$.

In other words, we do not expect the *known* upper bound due to GP to be tight, but instead expect that there exists some *unknown* bound that might be tighter. Notice that in the case where the bound by GP is as tight as the unknown bound by semi-metric ℓ in Assumption 2.1, our method still maintains an exponential convergence rate and an advantage over GP-UCB (no need for auxiliary optimization). Our method is expected to become relatively much better when the *known* bound due to GP is less tight compared to the unknown bound by ℓ.

As the semi-metric ℓ is unknown, there are infinitely many possible candidates that we can think of for ℓ. Accordingly, we simultaneously conduct global and local searches based on all the candidates of the bounds. The bound estimated by GP is used to reduce the number of candidates. Since the bound estimated by GP is known, we can ignore the candidates of the bounds that are looser than the bound estimated by GP. The source code of the proposed algorithm is publicly available at `http://lis.csail.mit.edu/code/imgpo.html`.

2.2.2 Description of Algorithm

Figure 2-1 illustrates how the algorithm works with a simple 1-dimensional objective function. We employ hierarchical partitioning to maintain hyperintervals, as illustrated by the line segments in the figure. We consider a hyperrectangle as our hyperinterval, with its center being the evaluation point of f (blue points in each line segment in Figure 2-1). For each iteration t, the algorithm performs the following

procedure *for each interval size*:

(i) Select the interval with the maximum center value among the intervals of the same size.

(ii) Keep the interval selected by (i) if it has a center value greater than that of any *larger* interval.

(iii) Keep the interval accepted by (ii) if it contains a UCB greater than the center value of any *smaller* interval.

(iv) If an interval is accepted by (iii), divide it along with the longest coordinate into three new intervals.

(v) For each new interval, if the UCB of the evaluation point is less than the best function value found so far, skip the evaluation and use the UCB value as the center value until the interval is accepted in step (ii) on some future iteration; otherwise, evaluate the center value.

(vi) Repeat steps (i)–(v) until every size of intervals are considered

Then, at the end of each iteration, the algorithm updates the GP hyperparameters. Here, the purpose of steps (i)–(iii) is to select an interval that might contain the global optimizer. Steps (i) and (ii) select the possible intervals based on the unknown bound by ℓ, while Step (iii) does so based on the bound by GP.

I now explain the procedure using the example in Figure 2-1. Let n be the number of divisions of intervals and let N be the number of function evaluations. t is the number of iterations. Initially, there is only one interval (the center of the input region $\Omega \subset \mathbb{R}$) and thus this interval is divided, resulting in the first diagram of Figure 2-1. At the beginning of iteration $t = 2$, step (i) selects the third interval from the left side in the first diagram ($t = 1, n = 2$), as its center value is the maximum. Because there are no intervals of different size at this point, steps (ii) and (iii) are skipped. Step (iv) divides the third interval, and then the GP hyperparameters are updated, resulting in the second diagram ($t = 2, n = 3$). At the beginning of iteration $t = 3$, it starts

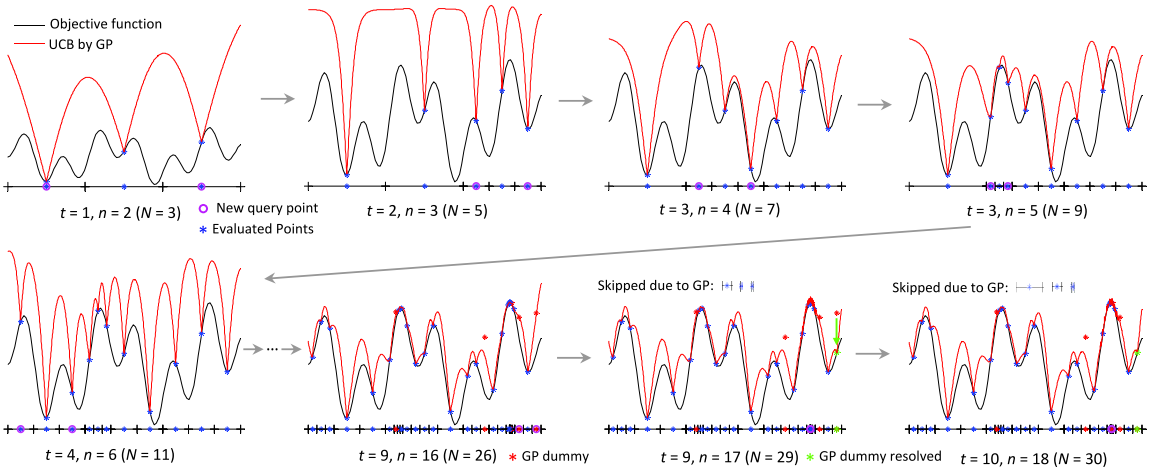


Figure 2-1: An illustration of IMGPO: t is the number of iteration, n is the number of divisions (or splits), N is the number of function evaluations.

conducting steps (i)–(v) for the largest intervals. Step (i) selects the second interval from the left side and step (ii) is skipped. Step (iii) accepts the second interval, because the UCB within this interval is no less than the center value of the smaller intervals, resulting in the third diagram ($t = 3, n = 4$). Iteration $t = 3$ continues by conducting steps (i)–(v) for the smaller intervals. Step (i) selects the second interval from the left side, step (ii) accepts it, and step (iii) is skipped, resulting in the forth diagram ($t = 3, n = 4$). The effect of the step (v) can be seen in the diagrams for iteration $t = 9$. At $n = 16$, the far right interval is divided, but no function evaluation occurs. Instead, UCB values given by GP are placed in the new intervals indicated by the red asterisks. One of the temporary dummy values is resolved at $n = 17$ when the interval is queried for division, as shown by the green asterisk. The effect of step (iii) for the rejection case is illustrated in the last diagram for iteration $t = 10$. At $n = 18$, t is increased to 10 from 9, meaning that the largest intervals are first considered for division. However, the three largest intervals are all rejected in step (iii), resulting in the division of a very small interval near the global optimum at $n = 18$.

2.2.3 Technical Detail of Algorithm

I define h to be the depth of the hierarchical partitioning tree, and $c_{h,i}$ to be the center point of the i^{th} hyperrectangle at depth h. N_{gp} is the number of the GP

Algorithm 2.1. Infinite-Metric GP Optimization (IMGPO)

Input: an objective function f, the search domain Ω, the GP kernel κ, $\Xi_{max} \in \mathbb{N}^+$ and $\eta \in (0,1)$

Initialize the set $\mathcal{T}_h = \{\emptyset\}\ \forall h \geq 0$
Set $c_{0,0}$ to be the center point of Ω and $\mathcal{T}_0 \leftarrow \{c_{0,0}\}$
Evaluate f at $c_{0,0}$: $g(c_{0,0}) \leftarrow f(c_{0,0})$
$f^+ \leftarrow g(c_{0,0}), \mathcal{D} \leftarrow \{(c_{0,0}, g(c_{0,0}))\}$
$n, N \leftarrow 1, N_{gp} \leftarrow 0, \Xi \leftarrow 1$
for $t = 1, 2, 3, ...$ **do**
  $\upsilon_{max} \leftarrow -\infty$
  for $h = 0$ to $depth(\mathcal{T})$ **do** # **for-loop for steps (i)-(ii)**
    while true **do**
      $i_h^* \leftarrow \arg\max_{i:c_{h,i} \in \mathcal{T}_h} g(c_{h,i})$
      if $g(c_{h,i_h^*}) < \upsilon_{max}$ **then**
        $i_h^* \leftarrow \emptyset$, **break**
      else if $g(c_{h,i_h^*})$ is *not* labeled as *GP-based* **then**
        $\upsilon_{max} \leftarrow g(c_{h,i_h^*})$, **break**
      else
        $g(c_{h,i_h^*}) \leftarrow f(c_{h,i_h^*})$ and remove the *GP-based* label from $g(c_{h,i_h^*})$
        $N \leftarrow N + 1,\ N_{gp} \leftarrow N_{gp} - 1$
        $\mathcal{D} \leftarrow \{\mathcal{D}, (c_{h,i_h^*}, g(c_{h,i_h^*}))\}$
  for $h = 0$ to $depth(\mathcal{T})$ **do** # **for-loop for step (iii)**
    if $i_h^* \neq \emptyset$ **then**
      $\xi \leftarrow$ the smallest positive integer s.t. $i_{h+\xi}^* \neq \emptyset$ and $\xi \leq \min(\Xi, \Xi_{max})$ if exists, and 0 otherwise
      $z(h, i_h^*) = \max_{k:c_{h+\xi,k} \in \mathcal{T}'_{h+\xi}(c_{h,i_h^*})} \mathcal{U}(c_{h+\xi,k}|\mathcal{D})$
      if $\xi \neq 0$ and $z(h, i_h^*) < g(c_{h+\xi,i_{h+\xi}^*})$ **then**
        $i_h^* \leftarrow \emptyset$, **break**
  $\upsilon_{max} \leftarrow -\infty$
  for $h = 0$ to $depth(\mathcal{T})$ **do** # **for-loop for steps (iv)-(v)**
    if $i_h^* \neq \emptyset$ and $g(c_{h,i_h^*}) \geq \upsilon_{max}$ **then**
      $n \leftarrow n + 1$.
      Divide the hyperrectangle centered at c_{h,i_h^*} along with the longest coordinate into three new hyperrectangles with the following centers:
      $$\mathcal{S} = \{c_{h+1,i(left)}, c_{h+1,i(center)}, c_{h+1,i(right)}\}$$
      $\mathcal{T}_{h+1} \leftarrow \{\mathcal{T}_{h+1}, \mathcal{S}\}$
      $\mathcal{T}_h \leftarrow \mathcal{T}_h \setminus c_{h,i_h^*},\ g(c_{h+1,i(center)}) \leftarrow g(c_{h,i_h^*})$
      for $i_{new} = \{i(left), i(right)\}$ **do**
        if $\mathcal{U}(c_{h+1,i_{new}}|\mathcal{D}) \geq f^+$ **then**
          $g(c_{h+1,i_{new}}) \leftarrow f(c_{h+1,i_{new}})$
          $\mathcal{D} \leftarrow \{\mathcal{D}, (c_{h+1,i_{new}}, g(c_{h+1,i_{new}}))\}$
          $N \leftarrow N + 1, f^+ \leftarrow \max(f^+, g(c_{h+1,i_{new}})),\ \upsilon_{max} = \max(\upsilon_{max}, g(c_{h+1,i_{new}}))$
        else
          $g(c_{h+1,i_{new}}) \leftarrow \mathcal{U}(c_{h+1,i_{new}}|\mathcal{D})$ and label $g(c_{h+1,i_{new}})$ as *GP-based*.
          $N_{gp} \leftarrow N_{gp} + 1$
  Update Ξ: if f^+ was updated, $\Xi \leftarrow \Xi + 2^2$, and otherwise, $\Xi \leftarrow \max(\Xi - 2^{-1}, 1)$
  Update GP hyperparameters by an empirical Bayesian method

evaluations. Define $depth(\mathcal{T})$ to be the largest integer h such that the set $\mathcal{T}_h$ is not empty. To compute UCB $\mathcal{U}$, I use $\varsigma_M = \sqrt{2\log(\pi^2 M^2/12\eta)}$ where M is the number of the calls made so far for $\mathcal{U}$ (i.e., each time we use $\mathcal{U}$, we increment M by one). This particular form of ς_M is to maintain the property of $f(x) \leq \mathcal{U}(x|\mathcal{D})$ during an execution of our algorithm with probability at least $1-\eta$. Here, η is the parameter of IMGPO. Ξ_{max} is another parameter, but it is only used to limit the possibly long computation of step (iii) (in the worst case, step (iii) computes UCBs $3^{\Xi_{max}}$ times although it would rarely happen).

The pseudocode is shown in Algorithm 2.1. Lines 8 to 23 correspond to steps (i)-(iii). These lines compute the index i_h^* of the candidate of the rectangle that may contain a global optimizer for each depth h. For each depth h, non-null index i_h^* at Line 24 indicates the remaining candidate of a rectangle that we want to divide. Lines 24 to 33 correspond to steps (iv)-(v) where the remaining candidates of the rectangles for all h are divided. To provide a simple executable division scheme (line 29), we assume Ω to be a hyperrectangle (see the last paragraph of section 4 for a general case).

Lines 8 to 17 correspond to steps (i)-(ii). Specifically, line 10 implements step (i) where a single candidate is selected for each depth, and lines 11 to 12 conduct step (ii) where some candidates are screened out. Lines 13 to 17 resolve the temporary dummy values computed by GP. Lines 18 to 23 correspond to step (iii) where the candidates are further screened out. At line 21, $\mathcal{T}'_{h+\xi}(c_{h,i_h^*})$ indicates the set of *all* center points of a fully expanded tree until depth $h+\xi$ *within* the region covered by the hyperrectangle centered at c_{h,i_h^*}. In other words, $\mathcal{T}'_{h+\xi}(c_{h,i_h^*})$ contains the nodes of the fully expanded tree rooted at c_{h,i_h^*} with depth ξ and can be computed by dividing the current rectangle at c_{h,i_h^*} and recursively divide all the resulting new rectangles until depth ξ (i.e., depth ξ from c_{h,i_h^*}, which is depth $h+\xi$ in the whole tree).

2.2.4 Relationship to Previous Algorithms

The most closely related algorithm is the BaMSOO algorithm [51], which combines SOO with GP-UCB. However, it only achieves a polynomial regret bound while

IMGPO achieves a exponential regret bound. IMGPO can achieve exponential regret because it utilizes the information encoded in the GP prior/posterior to reduce the degree of the ignorance of the unknown assumption with the semi-metric ℓ.

The idea of considering a set of infinitely many bounds was first proposed by Jones et al. [16]. Their DIRECT algorithm has been successfully applied to real-world problems [8, 55], but it only maintains the consistency property (i.e., convergence in the limit) from a theoretical viewpoint. DIRECT takes an input parameter ϵ to balance the global and local search efforts. This idea was generalized to the case of an unknown semi-metric and strengthened with a theoretical support (finite regret bound) by Munos [31] in the SOO algorithm. By limiting the depth of the search tree with a parameter h_{max}, the SOO algorithm achieves a finite regret bound that depends on *the near-optimality dimension*.

2.3 Analysis

In this section, I prove an exponential convergence rate of IMGPO and theoretically discuss the reason why the novel idea underling IMGPO is beneficial. The proofs are provided in the supplementary material. To examine the effect of considering infinitely many possible candidates of the bounds, I introduce the following term.

Definition 2.1. (Infinite-metric exploration loss). The infinite-metric exploration loss ρ_t is the number of intervals to be divided during iteration t.

The infinite-metric exploration loss ρ_τ can be computed as $\rho_t = \sum_{h=1}^{depth(\mathcal{T})} \mathbb{1}(i_h^* \neq \emptyset)$ at line 25. It is the cost (in terms of the number of function evaluations) incurred by not committing to any particular upper bound. If we were to rely on a specific bound, ρ_τ would be minimized to 1. For example, the DOO algorithm [31] has $\rho_t = 1\ \forall t \geq 1$. *Even if we know a particular upper bound*, relying on this knowledge and thus minimizing ρ_τ is not a good option *unless the known bound is tight enough compared to the unknown bound leveraged in our algorithm*. This will be clarified in our analysis. Let $\bar{\rho}_t$ be the maximum of the averages of $\rho_{1:t'}$ for $t' = 1, 2, ..., t$ (i.e.,

$\bar{\rho}_t \equiv \max(\{\frac{1}{t'}\sum_{\tau=1}^{t'}\rho_\tau\ ;\ t'=1,2,...,t\})$.

Assumption 2.2. For some pair of a global optimizer x^* and an *unknown* semi-metric ℓ that satisfies Assumption 1, both of the following, (i) shape on ℓ and (ii) lower bound constant, conditions hold:

(i) there exist $L>0$, $\alpha>0$ and $p\geq 1$ in $\mathbb{R}$ such that for all $x,x'\in\Omega$, $\ell(x',x)\leq L||x'-x||_p^\alpha$.
(ii) there exists $\theta\in(0,1)$ such that for all $x\in\Omega$, $f(x^*)\geq f(x)+\theta\ell\,(x,x^*)$.

In Theorem 2.1, I show that the exponential convergence rate $O\left(\lambda^{N+N_{gp}}\right)$ with $\lambda<1$ is achieved. I define $\Xi_n\leq\Xi_{max}$ to be the largest ξ used so far with n total node expansions. For simplicity, we assume that Ω is a square, which we satisfied in our experiments by scaling original Ω.

Theorem 2.1. Assume Assumptions 2.1 and 2.2. Let $\beta=\sup_{x,x'\in\Omega}\frac{1}{2}\|x-x'\|_\infty$. Let $\lambda=3^{-\frac{\alpha}{2CD\bar{\rho}_t}}<1$. Then, with probability at least $1-\eta$, the regret of IMGPO is bounded as

$$r_N\leq L(3\beta D^{1/p})^\alpha\exp\left(-\alpha\left[\frac{N+N_{gp}}{2CD\bar{\rho}_t}-\Xi_n-2\right]\ln 3\right)=O\left(\lambda^{N+N_{gp}}\right).$$

Importantly, our bound holds for the best values of the unknown L,α and p even though these values are not given. The closest result in previous work is that of BaMSOO [51], which obtained $\tilde{O}(n^{-\frac{2\alpha}{D(4-\alpha)}})$ with probability $1-\eta$ for $\alpha=\{1,2\}$. As can be seen, I have improved the regret bound. Additionally, in our analysis, we can see how L, p, and α affect the bound, allowing us to view the inherent difficulty of an objective function in a theoretical perspective. Here, C is a constant in N and is used in previous work [31, 51]. For example, if we conduct 2^D or 3^D-1 function evaluations per node-expansion and if $p=\infty$, we have that $C=1$.

We note that λ can get close to one as input dimension D increases, which suggests that there is a remaining challenge in scalability for higher dimensionality. One

strategy for addressing this problem would be to leverage additional assumptions such as those in [52, 17].

Remark 2.1. (The effect of the tightness of UCB by GP) If UCB computed by GP is "useful" such that $N/\bar{\rho}_t = \Omega(N)$, then our regret bound becomes

$$O\left(\exp\left(-\frac{N+N_{gp}}{2CD}\alpha\ln 3\right)\right).$$

If the bound due to UCB by GP is too loose (and thus useless), $\bar{\rho}_t$ can increase up to $O(N/t)$ (due to $\bar{\rho}_t \leq \sum_{i=1}^{t} i/t \leq O(N/t)$), resulting in the regret bound of $O\left(\exp\left(-\frac{t(1+N_{gp}/N)}{2CD}\alpha\ln 3\right)\right)$, which can be bounded by[1]

$$O\left(\exp\left(-\frac{N+N_{gp}}{2CD}\max(\frac{1}{\sqrt{N}},\frac{t}{N})\alpha\ln 3\right)\right).$$

Our proof works with this additional mechanism, but results in the regret bound with N being replaced by $\sqrt{N}$. Thus, if we assume to have at least "not useless" UCBs such that $N/\bar{\rho}_t = \Omega(\sqrt{N})$, this additional mechanism can be disadvantageous. Accordingly, we do not adopt it in our experiments.. This is still better than the known results.

Remark 2.2. (The effect of GP) Without the use of GP, our regret bound would be as follows: $r_N \leq L(3\beta D^{1/p})^{\alpha}\exp(-\alpha[\frac{N}{2CD}\frac{1}{\tilde{\rho}_t}-2]\ln 3)$, where $\bar{\rho}_t \leq \tilde{\rho}_t$ is the infinite-metric exploration loss without GP. Therefore, the use of GP reduces the regret bound by increasing N_{gp} and decreasing $\bar{\rho}_t$, but may potentially increase the bound by increasing $\Xi_n \leq \Xi$.

Remark 2.3. (The effect of infinite-metric optimization) To understand the effect of considering all the possible upper bounds, we consider the case without GP. If we consider all the possible bounds, we have the regret bound $L(3\beta D^{1/p})^{\alpha}\exp(-\alpha[\frac{N}{2CD}\frac{1}{\tilde{\rho}_t}-2]\ln 3)$ for *the best unknown* L, α and p. For standard optimization with a estimated

[1]This can be done by limiting the depth of search tree as $depth(T) = O(\sqrt{N})$.

Table 2.1: Average CPU time (in seconds) for the experiment with each test function

Algorithm	**S**in1	Sin2	Peaks	Rosenbrock2	Branin	Hartmann3	Hartmann6	Shekel5
GP-PI	29.66	115.90	47.90	921.82	1124.21	573.67	657.36	611.01
GP-EI	12.74	115.79	44.94	893.04	1153.49	562.08	604.93	558.58
SOO	0.19	0.19	0.24	0.744	0.33	0.30	0.25	0.29
BaMSOO	43.80	4.61	7.83	12.09	14.86	14.14	26.68	371.36
IMGPO	1.61	3.15	4.70	11.11	5.73	6.80	13.47	15.92

bound, we have $L'(3\beta D^{1/p'})^{\alpha'} \exp(-\alpha'[\frac{N}{2C'D} - 2] \ln 3)$ for an estimated L', α', and p'. By algebraic manipulation, considering all the possible bounds has a better regret when

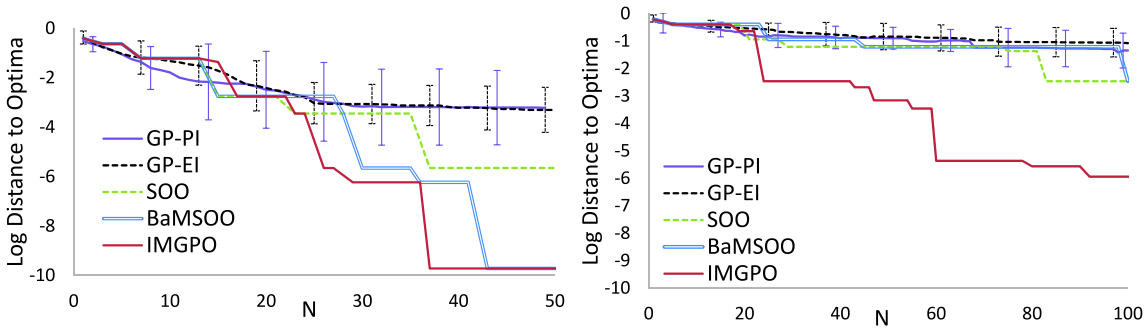
$$\tilde{\rho}_t^{-1} \geq \frac{2CD}{N \ln 3^{\alpha}}((\frac{N}{2C'D} - 2) \ln 3^{\alpha'} + 2 \ln 3^{\alpha} - \ln \frac{L'(3\beta D^{1/p'})^{\alpha'}}{L(3\beta D^{1/p})^{\alpha}}).$$

For an intuitive insight, we can simplify the above by assuming $\alpha' = \alpha$ and $C' = C$ as

$$\tilde{\rho}_t^{-1} \geq 1 - \frac{Cc_2 D}{N} \ln \frac{L' D^{\alpha/p'}}{L D^{\alpha/p}}.$$

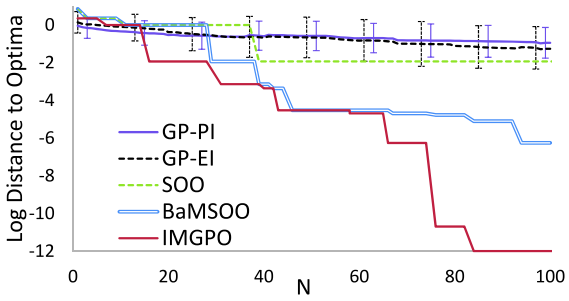
Because L and p are the ones that achieve the lowest bound, the logarithm on the right-hand side is always non-negative. Hence, $\tilde{\rho}_t = 1$ always satisfies the condition. When L' and p' are not tight enough, the logarithmic term increases in magnitude, allowing $\tilde{\rho}_t$ to increase. For example, if the second term on the right-hand side has a magnitude of greater than 0.5, then $\tilde{\rho}_t = 2$ satisfies the inequality. Therefore, even if we know the upper bound of the function, we can see that it may be better not to rely on this, but rather take the infinite many possibilities into account.

One may improve the algorithm with different division procedures than one presented in Algorithm 2.1. Accordingly, in the supplementary material, I derive an abstract version of the regret bound for IMGPO with a family of division procedures that satisfy some assumptions. This information could be used to design a new division procedure.

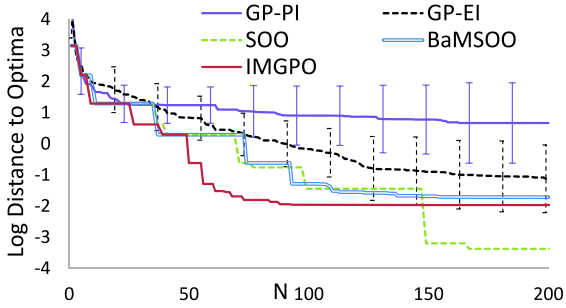


(a) Sin1: [1, 1.92, 2]

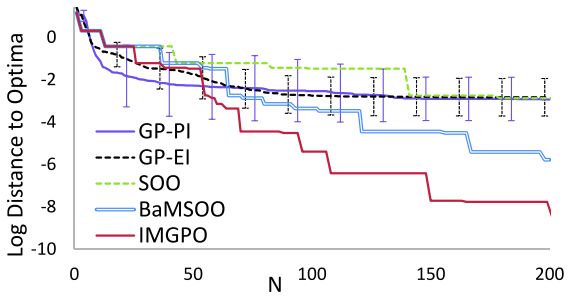
(b) Sin2: [2, 3.37, 3]



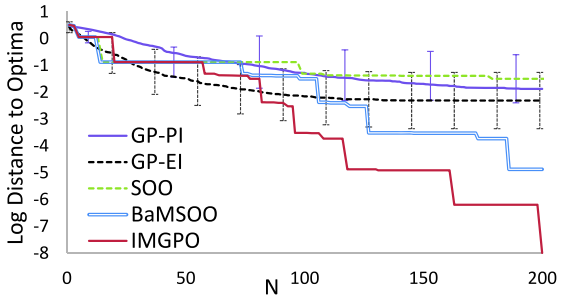
(c) Peaks: [2, 3.14, 4]



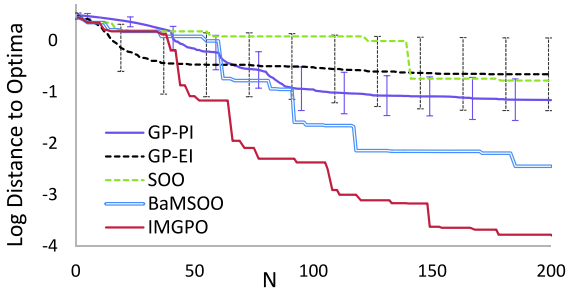
(d) Rosenbrock2: [2, 3.41, 4]



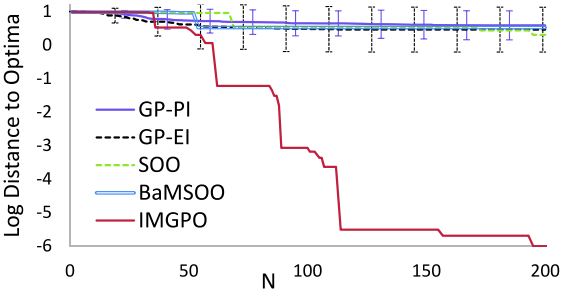
(e) Branin: [2, 4.44, 2]



(f) Hartmann3: [3, 4.11, 3]



(g) Hartmann6: [6, 4.39, 4]



(h) Shekel5: [4, 3.95, 4]

Figure 2-2: Performance Comparison: in the order, the digits inside of the parentheses [ ] indicate the dimensionality of each function, and the variables $\bar{\rho}_t$ and Ξ_n at the end of computation for IMGPO.

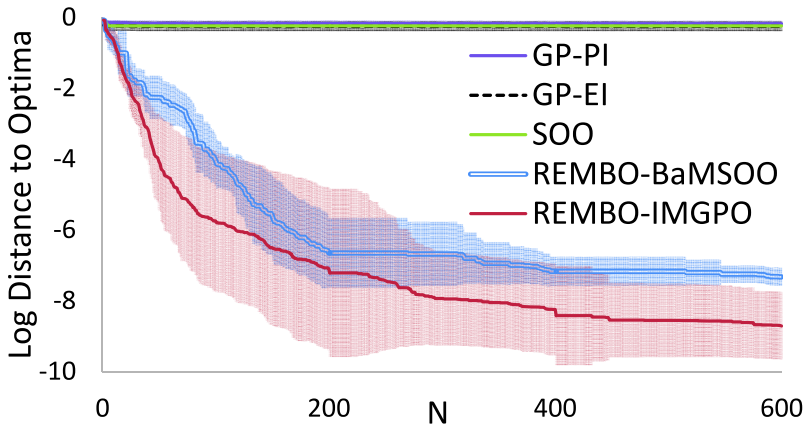


Figure 2-3: Sin1000: $[D = 1000, \bar{\rho} = 3.95, \Xi_n = 4]$

2.4 Experiments

In this section, I compare the IMGPO algorithm with the SOO, BaMSOO, GP-PI and GP-EI algorithms [31, 51, 39]. In previous work, BaMSOO and GP-UCB were tested with a pair of a handpicked good kernel and hyperparameters for each function [51]. In our experiments, we assume that the knowledge of good kernel and hyperparameters is unavailable, which is usually the case in practice. Thus, for IMGPO, BaMSOO, GP-PI and GP-EI, we simply used one of the most popular kernels, the isotropic Matern kernel with $\nu = 5/2$. This is given by $\kappa(x, x') = g(\sqrt{5||x - x'||^2/l})$, where $g(z) = \sigma^2(1 + z + z^2/3)\exp(-z)$. Then, I blindly initialized the hyperparameters to $\sigma = 1$ and $l = 0.25$ for all the experiments; these values were updated with an empirical Bayesian method after each iteration. To compute the UCB by GP, I used $\eta = 0.05$ for IMGPO and BaMSOO. For IMGPO, Ξ_{max} was fixed to be 2^2 (the effect of selecting different values is discussed later). For BaMSOO and SOO, the parameter h_{max} was set to $\sqrt{n}$, according to Corollary 4.3 in [31]. For GP-PI and GP-EI, I used the SOO algorithm and a local optimization method using gradients to solve the auxiliary optimization. For SOO, BaMSOO and IMGPO, I used the corresponding deterministic division procedure (given Ω, the initial point is fixed and no randomness exists). For GP-PI and GP-EI, I randomly initialized the first evaluation point and report the mean and one standard deviation for 50 runs.

The experimental results for eight different objective functions are shown in Figure

2-2 and 2-3. The vertical axis is $\log_{10}(f(x^*)-f(x^+))$, where $f(x^*)$ is the global optima and $f(x^+)$ is the best value found by the algorithm. Hence, the lower the plotted value on the vertical axis, the better the algorithm's performance. The last five functions are standard benchmarks for global optimization [48]. The first two were used in [31] to test SOO, and can be written as $f_{sin1}(x) = (\sin(13x)\sin+1)/2$ for Sin1 and $f_{sin2}(x) = f_{sin1}(x_1)f_{sin1}(x_2)$ for Sin2. The form of the third function is given in Equation (16) and Figure 2 in [29]. The last function in Figure 2-3 is Sin2 embedded in 1000 dimension in the same manner described in Section 4.1 in [52], which is used here to illustrate a possibility of using IMGPO as a main subroutine to scale up to higher dimensions with additional assumptions. For this function, I used REMBO [52] with IMGPO and BaMSOO as its Bayesian optimization subroutine. All of these functions are multimodal, except for Rosenbrock2, with dimensionality from 1 to 1000.

As we can see from Figure 2-2 and 2-3, IMGPO outperformed the other algorithms in general. SOO produced the competitive results for Rosenbrock2 because our GP prior was misleading (i.e., it did not model the objective function well and thus the property $f(x) \leq \mathcal{U}(x|\mathcal{D})$ did not hold many times). As can be seen in Table 2.1, IMGPO is much faster than traditional GP optimization methods although it is slower than SOO. For Sin 1, Sin2, Branin and Hartmann3, increasing Ξ_{max} does not affect IMGPO because Ξ_n did not reach $\Xi_{max} = 2^2$ (Figure 2-2 and 2-3). For the rest of the test functions, we would be able to improve the performance of IMGPO by increasing Ξ_{max} at the cost of extra CPU time.

Chapter 3

Bounded Optimal Exploration in MDP

In the previous chapter, we discussed a solution for the problem of having a loose bound by taking advantage of the natural existence of *tighter* yet *unknown* bounds. In this chapter, we consider another approach for a different member of bound-based methods.

Within the framework of probably approximately correct Markov Decision Processes (PAC-MDP), much theoretical work has focused on methods to attain near optimality after a relatively long period of learning and exploration. However, practical concerns require the attainment of satisfactory behavior within a short period of time. In this chapter, I relax the PAC-MDP conditions to reconcile theoretically driven exploration methods and practical needs. I propose simple algorithms for discrete and continuous state spaces, and illustrate the benefits of our proposed relaxation via theoretical analyses and numerical examples. Our algorithms also maintain anytime error bounds and average loss bounds. Our approach accommodates both Bayesian and non-Bayesian methods.

3.1 Preliminaries

An MDP [34] can be represented as a tuple (S, A, R, P, γ), where S is a set of states, A is a set of actions, P is the transition probability function, R is a reward function,

and γ is a discount factor. The value of policy π at state s, $V^{\pi}(s)$, is the cumulative (discounted) expected reward, which is given by

$$V^{\pi}(s) = E\left[\sum_{i=0}^{\infty} \gamma^{i} R\left(s_i, \pi(s_i), s_{i+1}\right) \mid s_0 = s, \pi\right],$$

where the expectation is over the sequence of states $s_{i+1} \sim P(S|s_i, \pi(s_i))$ for all $i \geq 0$. Using Bellman's equation, the value of the optimal policy or the optimal value, $V^{*}(s)$, can be written as $V^{*}(s) = \max_a \sum_{s'} P(s'|s,a))[R(s,a,s') + \gamma V^{*}(s')]$.

In many situations, the transition function P and/or the reward function R are initially unknown. Under such conditions, we often want a policy of an algorithm at time t, $\mathcal{A}_t$, to yield a value $V^{\mathcal{A}_t}(s_t)$ that is close to the optimal value $V^{*}(s_t)$ after some exploration. Here, s_t denotes the current state at time t. More precisely, we may want the following: for all $\epsilon > 0$ and for all $\delta = (0, 1)$, $V^{\mathcal{A}_t}(s_t) \geq V^{*}(s_t) - \epsilon$, with probability at least $1 - \delta$ when $t \geq \tau$, where τ is the exploration time. The algorithm with a policy$\mathcal{A}_t$ is said to be "probably approximately correct" for MDPs (PAC-MDP) [42] if this condition holds with τ being at most polynomial in the relevant quantities of MDPs. The notion of PAC-MDP has a strong theoretical basis and is widely applicable, avoiding the need for additional assumptions, such as reachability in state space [15], access to a reset action [13], and access to a parallel sampling oracle [21].

However, the PAC-MDP approach often results in an algorithm over-exploring the state space, causing a low reward per unit time for a long period of time. Accordingly, past studies that proposed PAC-MDP algorithms have rarely presented a corresponding experimental result, or have done so by tuning the free parameters, which renders the relevant algorithm no longer PAC-MDP [45, 23, 40]. This problem was noted in [23, 6, 19]. Furthermore, in many problems, it may not even be possible to guarantee $V^{\mathcal{A}_t}$ close to V^{*} within the agent's lifetime. Li [26] noted that, despite the strong theoretical basis of the PAC-MDP approach, heuristic-based methods remain popular in practice. This would appear to be a result of the above issues. In summary, there seems to be a dissonance between a strong theoretical approach and practical needs.

3.2 Bounded Optimal Learning

The practical limitations of the PAC-MDP approach lie in their focus on correctness without accommodating the time constraints that occur naturally in practice. To overcome the limitation, we first define the notion of *reachability in model learning*, and then relax the PAC-MDP objective based on it. For brevity, we focus on the transition model.

3.2.1 Reachability in Model Learning

For each state-action pair (s, a), let $M_{(s,a)}$ be a set of all transition models and $\widehat{P}_t(\cdot|s,a) \in M_{(s,a)}$ be the current model at time t (i.e., $\widehat{P}_t(\cdot|s,a) : S \to [0,\infty)$). Define $S'_{(s,a)}$ to be a set of possible future samples as $S'_{(s,a)} = \{s'|P(s'|s,a) > 0\}$. Let $f_{(s,a)} : M_{(s,a)} \times S'_{(s,a)} \to M_{(s,a)}$ represent the model update rule; $f_{(s,a)}$ maps a model (in $M_{(s,a)}$) and a new sample (in $S'_{(s,a)}$) to a corresponding new model (in $M_{(s,a)}$). We can then write $\mathcal{L} = (M, f)$ to represent a learning method of an algorithm, where $M = \cup_{(s,a)\in(S,A)} M_{(s,a)}$ and $f = \{f_{(s,a)}\}_{(s,a)\in(S,A)}$.

The set of h-reachable models, $\mathcal{M}_{\mathcal{L},t,h,(s,a)}$, is recursively defined as $\mathcal{M}_{\mathcal{L},t,h,(s,a)} = \{\widehat{P}' \in M_{(s,a)} | \widehat{P}' = f_{(s,a)}(\widehat{P}, s') \text{ for some } \widehat{P} \in \mathcal{M}_{\mathcal{L},t,h-1,(s,a)} \text{ and } s' \in S'_{(s,a)}\}$ with the boundary condition $\mathcal{M}_{t,0,(s,a)} = \{\widehat{P}_t(\cdot|s,a)\}$.

Intuitively, the set of h-reachable models, $\mathcal{M}_{\mathcal{L},t,h,(s,a)} \subseteq M_{(s,a)}$, contains the transition models that can be obtained if the agent updates the current model at time t using any combination of h additional samples $s'_1, s'_2, ..., s'_h \sim P(S|s,a)$. Note that the set of h-reachable models is defined *separately for each state-action pair*. For example, $\mathcal{M}_{\mathcal{L},t,h,(s_1,a_1)}$ contains only those models that are reachable using the h additional samples drawn from $P(S|s_1,a_1)$.

We define the h-reachable optimal value $V^{d*}_{\mathcal{L},t,h}(s)$ with respect to a distance function d as

$$V^{d*}_{\mathcal{L},t,h}(s) = \max_a \sum_{s'} \widehat{P}^{d*}_{\mathcal{L},t,h}(s'|s,a)[R(s,a,s') + \gamma V^{d*}_{\mathcal{L},t,h}(s')],$$

where

$$\widehat{P}^{d*}_{\mathcal{L},t,h}(\cdot|s,a) = \underset{\widehat{P}\in\mathcal{M}_{\mathcal{L},t,h,(s,a)}}{\arg\min} \quad d(\widehat{P}(\cdot|s,a), P(\cdot|s,a)).$$

Intuitively, the h-reachable optimal value, $V^{d*}_{\mathcal{L},t,h}(s)$, is the optimal value estimated with the "best" model in the set of h-reachable models (here, the term "best" is in terms of the distance function $d(\cdot,\cdot)$).

3.2.2 PAC in Reachable MDP

Using the concept of reachability in model learning, we define the notion of "probably approximately correct" in an h-reachable MDP (PAC-RMDP(h)). Let $\mathcal{P}(x_1, x_2, ..., x_n)$ be a polynomial in $x_1, x_2, ..., x_n$ and |MDP| be the complexity of an MDP [26].

Definition 3.1. (PAC-RMDP(h)) An algorithm with a policy $\mathcal{A}_t$ and a learning method $\mathcal{L}$ is PAC-RMDP(h) with respect to a distance function d if for all $\epsilon > 0$ and for all $\delta = (0, 1)$,

1) there exists $\tau = O(\mathcal{P}(1/\epsilon, 1/\delta, 1/(1-\gamma), |\text{MDP}|, h))$ such that for all $t \geq \tau$,

$$V^{\mathcal{A}_t}(s_t) \geq V^{d*}_{\mathcal{L},t,h}(s_t) - \epsilon$$

with probability at least $1 - \delta$, *and*

2) there exists $h^*(\epsilon, \delta) = O(\mathcal{P}(1/\epsilon, 1/\delta, 1/(1-\gamma), |\text{MDP}|))$ such that for all $t \geq 0$,

$$|V^*(s_t) - V^{d*}_{\mathcal{L},t,h^*(\epsilon,\delta)}(s_t)| \leq \epsilon.$$

with probability at least $1 - \delta$.

The first condition ensures that the agent efficiently learns the h-reachable models. The second condition guarantees that the learning method $\mathcal{L}$ and the distance function d are not arbitrarily poor.

In the following, we relate PAC-RMDP(h) to PAC-MDP and near-Bayes optimality. The proofs are given in the appendix at the end of this thesis.

Proposition 3.1. (PAC-MDP) If an algorithm is PAC-RMDP($h^*(\epsilon, \delta)$), then it is PAC-MDP, where $h^*(\epsilon, \delta)$ is given in Definition 3.1.

Proposition 3.2. (Near-Bayes optimality) Consider model-based Bayesian reinforcement learning [46]. Let H be a planning horizon in the belief space b. Assume that the Bayesian optimal value function, $V^*_{b,H}$, converges to the H-reachable optimal function such that, for all $\epsilon > 0$, $|V^{d*}_{\mathcal{L},t,H}(s_t) - V^*_{b,H}(s_t, b_t)| \leq \epsilon$ for all but polynomial time steps. Then, a PAC-RMDP(H) algorithm with a policy $\mathcal{A}_t$ obtains an expected cumulative reward $V^{\mathcal{A}_t}(s_t) \geq V^*_{b,H}(s_t, b_t) - 2\epsilon$ for all but polynomial time steps with probability at least $1 - \delta$.

Note that $V^{\mathcal{A}_t}(s_t)$ is the *actual* expected cumulative reward with the expectation over the true dynamics P, whereas $V^*_{b,H}(s_t, b_t)$ is the *believed* expected cumulative reward with the expectation over the current belief b_t and its belief evolution. In addition, whereas the PAC-RMDP(H) condition guarantees convergence to an H-reachable optimal value function, Bayesian optimality does *not*[1]. In this sense, Proposition 3.2 suggests that the theoretical guarantee of PAC-RMDP(H) would be stronger than that of near-Bayes optimality with an H step lookahead.

Summarizing the above, PAC-RMDP($h^*(\epsilon, \delta)$) implies PAC-MDP, and PAC-RMD P(H) is related to near-Bayes optimality. Moreover, as h decreases in the range $(0, h^*)$ or $(0, H)$, the theoretical guarantee of PAC-RMDP(h) becomes weaker than previous theoretical objectives. This accommodates the practical need to improve the trade-off between the theoretical guarantee (i.e., optimal behavior after a long period of exploration) and practical performance (i.e., satisfactory behavior after a reasonable period of exploration) via the concept of reachability. We discuss the relationship to

[1] A Bayesian estimation with random samples converges to the true value under certain assumptions. However, for exploration, the selection of actions can cause the Bayesian optimal agent to ignore some state-action pairs, removing the guarantee of the convergence. This effect was well illustrated by Li (2009, Example 9).

bounded rationality [38] and bounded optimality [36] as well as the corresponding notions of regret and average loss in the appendix.

3.3 Discrete Domain

To illustrate the proposed concept, we first consider a simple case involving finite state and action spaces with an unknown transition function P. Without loss of generality, we assume that the reward function R is known.

3.3.1 Algorithm

Let $\tilde{V}^{\mathcal{A}}(s)$ be the internal value function used by the algorithm to choose an action. Let $V^{\mathcal{A}}(s)$ be the actual value function according to true dynamics P. To derive the algorithm, we use the principle of optimism in the face of uncertainty, such that $\tilde{V}^{\mathcal{A}}(s) \geq V^{d*}_{\mathcal{L},t,h}(s)$ for all $s \in S$. This can be achieved using the following internal value function:

$$\tilde{V}^{\mathcal{A}}(s) = \max_{\substack{a, \\ \tilde{P} \in \mathcal{M}_{\mathcal{L},t,h,(s,a)}}} \sum_{s'} \tilde{P}(s'|s,a)[R(s,a,s') + \gamma \tilde{V}^{\mathcal{A}}(s')] \tag{3.1}$$

The pseudocode is shown in Algorithm 3.1. In the following, we consider the special case in which we use the sample mean estimator (which determines $\mathcal{L}$). That is, we use $\widehat{P}_t(s'|s,a) = n_t(s,a,s')/n_t(s,a)$, where $n_t(s,a)$ is the number of samples for the state-action pair (s,a), and $n_t(s,a,s')$ is the number of samples for the transition from s to s' given an action a. In this case, the maximum over the model in Equation (3.1) is achieved when all future h observations are transitions to the state with the best value. Thus, $\tilde{V}^{\mathcal{A}}$ can be computed by $\tilde{V}^{\mathcal{A}}(s) = \max_a \sum_{s' \in S} \frac{n_t(s,a,s')}{n_t(s,a)+h}[R(s,a,s') + \gamma \tilde{V}^{\mathcal{A}}(s')] + \max_{s'} \frac{h}{n_t(s,a)+h}[R(s,a,s') + \gamma \tilde{V}^{\mathcal{A}}(s')]$.

Algorithm 3.1. Discrete PAC-RMDP

Input: $h \geq 0$

for time step $t = 1, 2, 3, ...$ **do**
Action: Take action based on $\tilde{V}^A(s_t)$ in Equation (3.1)
Observation: Save the sufficient statistics
Estimate: Update the model $\widehat{P}_{t,0}$

3.3.2 Analysis

We first show that Algorithm 3.1 is PAC-RMDP(h) for all $h \geq 0$ (Theorem 3.1), maintains an anytime error bound and average loss bound (Corollary 3.1 and the following discussion), and is related with previous algorithms (Remarks 3.1 and 3.2). We then analyze its *explicit exploration runtime* (Definition 3.3). We assume that Algorithm 3.1 is used with the sample mean estimator, which determines $\mathcal{L}$. We fix the distance function as $d(\widehat{P}(\cdot|s,a), P(\cdot|s,a)) = \|\widehat{P}(\cdot|s,a) - P(\cdot|s,a)\|_1$. The proofs are given in the appendix.

Theorem 3.1. (PAC-RMDP) Let $\mathcal{A}_t$ be a policy of Algorithm 3.1. Let

$$z = \max(h, \frac{\ln(2^{|S|}|S||A|/\delta)}{\epsilon(1-\gamma)}).$$

Then, for all $\epsilon > 0$, for all $\delta = (0, 1)$, and for all $h \geq 0$,

1) for all but at most $O\left(\frac{z|S||A|}{\epsilon^2(1-\gamma)^2} \ln \frac{|S||A|}{\delta}\right)$ time steps, $V^{\mathcal{A}_t}(s_t) \geq V^{d*}_{\mathcal{L},t,h}(s_t) - \epsilon$, with probability at least $1 - \delta$, *and*

2) there exist $h^*(\epsilon, \delta) = O(\mathcal{P}(1/\epsilon, 1/\delta, 1/(1-\gamma), |\text{MDP}|))$ such that

$$|V^*(s_t) - V^{d*}_{\mathcal{L},t,h^*(\epsilon,\delta)}(s_t)| \leq \epsilon$$

with probability at least $1 - \delta$.

Definition 3.2. (Anytime error) The anytime error $\epsilon_{t,h} \in \mathbb{R}$ is the smallest value such that $V^{\mathcal{A}_t}(s_t) \geq V^{d*}_{\mathcal{L},t,h}(s_t) - \epsilon_{t,h}$.

Corollary 3.1. (Anytime error bound) With probability at least $1-\delta$, if $h \leq \frac{\ln(2^{|S|}|S||A|/\delta)}{\epsilon(1-\gamma)}$,

$$\epsilon_{t,h} = O\left(\sqrt[3]{\frac{|S||A|}{t(1-\gamma)^3}\ln\frac{|S||A|}{\delta}\ln\frac{2^{|S|}|S||A|}{\delta}}\right);$$

otherwise,

$$\epsilon_{t,h} = O\left(\sqrt{\frac{h|S||A|}{t(1-\gamma)^2}\ln\frac{|S||A|}{\delta}}\right).$$

The anytime T-step average loss is equal to $\frac{1}{T}\sum_{t=1}^{T}(1-\gamma^{T+1-t})\epsilon_{t,h}$. Moreover, in this simple problem, we can relate Algorithm 3.1 to a particular PAC-MDP algorithm and a near-Bayes optimal algorithm.

Remark 3.1. (Relation to MBIE) Let $m = O(\frac{|S|}{\epsilon^2(1-\gamma)^4} + \frac{1}{\epsilon^2(1-\gamma)^4}\ln\frac{|S||A|}{\epsilon(1-\gamma)\delta})$. Let $h^*(s,a) = \frac{n(s,a)z(s,a)}{1-z(s,a)}$, where $z(s,a) = 2\sqrt{2[\ln(2^{|S|}-2) - \ln(\delta/(2|S||A|m))]/n(s,a)}$. Then, Algorithm 3.1 with the input parameter $h = h^*(s,a)$ behaves identically to a PAC-MDP algorithm, Model Based Interval Estimation (MBIE) [43], the sample complexity of which is $O(\frac{|S||A|}{\epsilon^3(1-\gamma)^6}(|S| + \ln\frac{|S||A|}{\epsilon(1-\gamma)\delta})\ln\frac{1}{\delta}\ln\frac{1}{\epsilon(1-\gamma)})$.

Remark 3.2. (Relation to BOLT) Let $h = H$, where H is a planning horizon in the belief space b. Assume that Algorithm 3.1 is used with an independent Dirichlet model for each (s,a), which determines $\mathcal{L}$. Then, Algorithm 3.1 behaves identically to a near-Bayes optimal algorithm, Bayesian Optimistic Local Transitions (BOLT) [3], the sample complexity of which is $O(\frac{H^2|S||A|}{\epsilon^2(1-\gamma)^2}\ln\frac{|S||A|}{\delta})$.

As expected, the sample complexity for PAC-RMDP(h) (Theorem 3.1) is smaller than that for PAC-MDP (Remark 3.1) (at least when $h \leq |S|(1-\gamma)^{-3}$), but larger than that for near-Bayes optimality (Remark 3.2) (at least when $h \geq H$). Note that BOLT is not necessarily PAC-RMDP(h), because misleading priors can violate both conditions in Definition 3.1.

Further Discussion

An important observation is that, when $h \leq \frac{|S|}{\epsilon(1-\gamma)} \ln \frac{|S||A|}{\delta}$, the sample complexity of Algorithm 3.1 is dominated by the number of samples required to refine the model, rather than the explicit exploration of unknown aspects of the world. Recall that the internal value function $\tilde{V}^{\mathcal{A}}$ is designed to force the agent to explore, whereas the use of the currently estimated value function $V^{d*}_{\mathcal{L},t,0}(s)$ results in exploitation. The difference between $\tilde{V}^{\mathcal{A}}$ and $V^{*}_{\mathcal{L},t,0}(s)$ decreases at a rate of $O(h/n_t(s,a))$, whereas the error between $V^{\mathcal{A}}$ and $V^{d*}_{\mathcal{L},t,0}(s)$ decreases at a rate of $O(1/\sqrt{n_t(s,a)})$. Thus, Algorithm 3.1 would stop the explicit exploration much sooner (when $\tilde{V}^{\mathcal{A}}$ and $V^{d*}_{\mathcal{L},t,0}(s)$ become close), and begin exploiting the model, while still refining it, so that $V^{d*}_{\mathcal{L},t,0}(s)$ tends to $V^{\mathcal{A}}$. In contrast, PAC-MDP algorithms are forced to explore until the error between $V^{\mathcal{A}}$ and V^{*} becomes sufficiently small, where the error decreases at a rate of $O(1/\sqrt{n_t(s,a)})$. This provides some intuition to explain why a PAC-RMDP(h) algorithm with small h may avoid over-exploration, and yet, in some cases, learn the true dynamics to a reasonable degree, as shown in the experimental examples.

In the following, we formalize the above discussion.

Definition 3.3. (Explicit exploration runtime) The *explicit exploration runtime* is the smallest integer τ such that for all $t \geq \tau$, $|\tilde{V}^{\mathcal{A}_t}(s_t) - V^{d*}_{\mathcal{L},t,0}(s_t)| \leq \epsilon$.

Corollary 3.2. (Explicit exploration bound) With probability at least $1-\delta$, the explicit exploration runtime of Algorithm 3.1 is

$$O(\frac{h|S||A|}{\epsilon(1-\gamma)\Pr[A_K]} \ln \frac{|S||A|}{\delta}) = O(\frac{h|S||A|}{\epsilon^2(1-\gamma)^2} \ln \frac{|S||A|}{\delta}),$$

where A_K is the escape event defined in the proof of Theorem 3.1.

If we assume $\Pr[A_K]$ to stay larger than a fixed constant, or to be very small ($\leq \frac{\epsilon(1-\gamma)}{3R_{max}}$) (so that $\Pr[A_K]$ does not appear in Corollary 3.2 as shown in the corresponding case analysis for Theorem 3.1), the explicit exploration runtime can be reduced to $O(\frac{h|S||A|}{\epsilon(1-\gamma)} \ln \frac{|S||A|}{\delta})$. Intuitively, this happens when the given MDP does not have low

yet not-too low probability and high-consequence transition that is initially unknown. Naturally, such a MDP is difficult to learn, as reflected in Corollary 3.2.

3.3.3 Experimental Example

We compare the proposed algorithm with MBIE [43], variance-based exploration (VBE) [40], Bayesian Exploration Bonus (BEB) [23], and BOLT [3]. These algorithms were designed to be PAC-MDP or near-Bayes optimal, but have been used with parameter settings that render them neither PAC-MDP nor near-Bayes optimal. In contrast to the experiments in previous research, we present results with ϵ set to several theoretically meaningful values[2] as well as one theoretically non-meaningful value to illustrate its property[3]. Because our algorithm is deterministic with no sampling and no assumptions on the input distribution, we do not compare it with algorithms that use sampling, or rely heavily on knowledge of the input distribution.

We consider a five-state chain problem [46], which is a standard toy problem in the literature. In this problem, the optimal policy is to move toward the state farthest from the initial state, but the reward structure explicitly encourages an exploitation agent, or even an ϵ-greedy agent, to remain in the initial state. We use a discount factor of $\gamma = 0.95$ and a convergence criterion for the value iteration of $\epsilon' = 0.01$.

Figure 3-1 shows the numerical results in terms of the average reward per time step (average over 1000 runs). As can be seen from this figure, the proposed algorithm worked better. MBIE and VBE work reasonably if we discard the theoretical guarantee. As the maximum reward is $R_{max} = 1$, the upper bound on the value function is $\sum_{i=1}^{\infty} \gamma^i R_{max} = \frac{1}{1-\gamma} R_{max} = 20$. Thus, ϵ-closeness does not yield any useful information when $\epsilon \geq 20$. A similar problem was noted by Kolter and Ng [23] and

[2]MBIE is PAC-MDP with the parameters δ and ϵ. VBE is PAC-MDP in the assumed (prior) input distribution with the parameter δ. BEB and BOLT are near-Bayes optimal algorithms whose parameters β and η are fully specified by their analyses, namely $\beta = 2H^2$ and $\eta = H$. Following Araya-López et al. [3], we set β and η using the ϵ'-approximated horizon $H \approx \lceil \log_\gamma(\epsilon'(1-\gamma)) \rceil = 148$. We use the sample mean estimator for the PAC-MDP and PAC-RMDP(h) algorithms, and an independent Dirichlet model for the near-Bayes optimal algorithms.

[3]We can interpolate their qualitative behaviors with values of ϵ other than those presented here. This is because the principle behind our results is that small values of ϵ causes over-exploration due to the focus on the near-optimality.

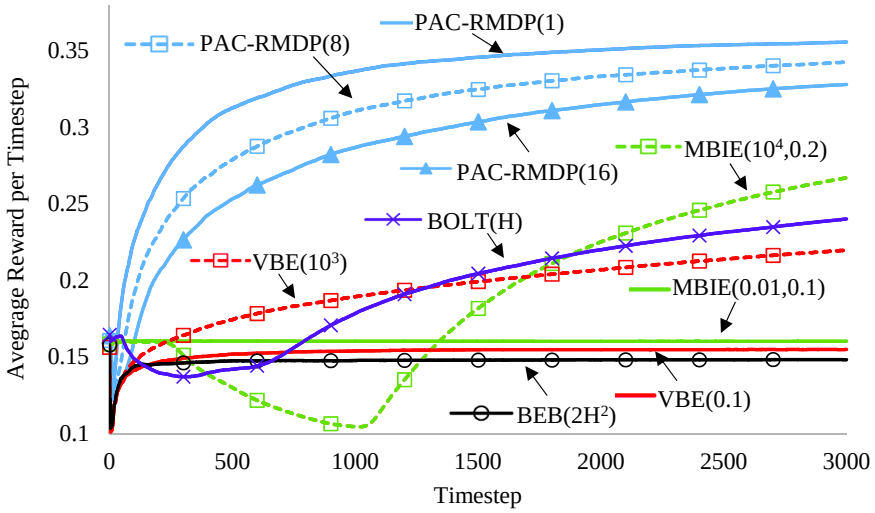


Figure 3-1: Average total reward per time step for the Chain Problem. The algorithm parameters are shown as PAC-RMDP(h), MBIE(ϵ, δ), VBE(δ), BEB(β), and BOLT(η).

Araya-López et al. [3].

In the appendix, we present the results for a problem with low-probability high-consequence transitions, in which PAC-RMDP(8) produced the best result.

3.4 Continuous Domain

In this section, we consider the problem of a continuous state space and discrete action space. The transition function is possibly nonlinear, but can be linearly parameterized as: $s_{t+1}^{(i)} = \theta_{(i)}^T \Phi_{(i)}(s_t, a_t) + \zeta_t^{(i)}$, where the state $s_t \in S \subseteq \mathbb{R}^{n_S}$ is represented by n_S state parameters ($s^{(i)} \in \mathbb{R}$ with $i \in \{1, ..., n_s\}$), and $a_t \in A$ is the action at time t. We assume that the basis functions $\Phi_{(i)} : S \times A \rightarrow \mathbb{R}^{n_i}$ are known, but the weights $\theta \in \mathbb{R}^{n_i}$ are unknown. $\zeta_t^{(i)} \in \mathbb{R}$ is the noise term and given by $\zeta_t^{(i)} \sim \mathcal{N}(0, \sigma_{(i)}^2)$. In other words, $P(s_{t+1}^{(i)} | s_t, a_t) = \mathcal{N}(\theta_{(i)}^T \Phi_{(i)}(s_t, a_t), \sigma_{(i)}^2)$. For brevity, we focus on unknown transition dynamics, but our method is directly applicable to unknown reward functions if the reward is represented in the above form. This problem is a slightly generalized version of those considered by Abbeel and Ng [1], Strehl and Littman [44], and Li et al. [27].

3.4.1 Algorithm

We first define the variables used in our algorithm, and then explain how the algorithm works. Let $\hat{\theta}_{(i)}$ be the vector of the model parameters for the i^{th} state component. Let $X_{t,i} \in \mathbb{R}^{t \times n_i}$ consist of t input vectors $\Phi^T_{(i)}(s,a) \in \mathbb{R}^{1 \times n_i}$ at time t. We then denote the eigenvalue decomposition of the input matrix as $X^T_{t,i}X_{t,i} = U_{t,i}D_{t,i}(\lambda_{(1)}, \ldots, \lambda_{(n)})U^T_{t,i}$, where $D_{t,i}(\lambda_{(1)}, ..., \lambda_{(n)}) \in \mathbb{R}^{n_i \times n_i}$ represents a diagonal matrix. For simplicity of notation, we arrange the eigenvectors and eigenvalues such that the diagonal elements of $D_{t,i}(\lambda_{(1)}, ..., \lambda_{(n)})$ are $\lambda_{(1)}, ..., \lambda_{(j)} \geq 1$ and $\lambda_{(j+1)}, ..., \lambda_{(n)} < 1$ for some $0 \leq j \leq n$. We now define the main variables used in our algorithm: $z_{t,i} := (X^T_{t,i}X_{t,i})^{-1}, g_{t,i} := U_{t,i}D_{t,i}(\frac{1}{\lambda_{(1)}}, \ldots, \frac{1}{\lambda_{(j)}}, 0, \ldots, 0)U^T_{t,i}$, and $w_{t,i} := U_{t,i}D_{t,i}(0, \ldots, 0, 1_{(j+1)}, \ldots, 1_{(n)})U^T_{t,i}$. Let $\Delta^{(i)} \geq \sup_{s,a}|(\theta_{(i)} - \hat{\theta}_{(i)})^T\Phi_{(i)}(s,a)|$ be the upper bound on the model error. Define $\varsigma(M) = \sqrt{2\ln(\pi^2 M^2 n_s h/(6\delta))}$ where M is the number of calls for $\mathbf{I}_h$ (i.e., the number of computing $\tilde{r}$ in Algorithm 3.2).

With the above variables, we define the h-reachable model interval I_h as

$$
\begin{aligned}
&I_h(\Phi_{(i)}(s,a), X_{t,i})/[h(\Delta^{(i)} + \varsigma(M)\sigma_{(i)})] \\
&= |\Phi^T_{(i)}(s,a)g_{t,i}\Phi_{(i)}(s,a)| + \|\Phi^T_{(i)}(s,a)z_{t,i}\|\|w_{t,i}\Phi_{(i)}(s,a)\|.
\end{aligned}
$$

The h-reachable model interval is a function that maps a new state-action pair considered in the planning phase, $\Phi_{(i)}(s,a)$, and the agent's experience, $X_{t,i}$, to the upper bound of the error in the model prediction. We define the column vector consisting of n_S h-reachable intervals as $\mathbf{I}_h(s,a,X_t) = [I_h(\Phi_{(1)}(s,a), X_{t,1}), ..., I_h(\Phi_{(n_S)}(s,a), X_{t,n_S})]^T$.

We also leverage the continuity of the internal value function $\tilde{V}$ to avoid an expensive computation (to translate the error in the model to the error in value).

Assumption 3.1. (Continuity) There exists $L \in \mathbb{R}$ such that, for all $s, s' \in S$, $|\tilde{V}^*(s) - \tilde{V}^*(s')| \leq L\|s - s'\|$.

We set the degree of optimism for a state-action pair to be proportional to the uncertainty of the associated model. Using the h-reachable model interval, this can be achieved by simply adding a reward bonus that is proportional to the interval.

Algorithm 3.2. Linear PAC-RMDP

Input: h, δ Optional: $\Delta^{(i)}, L$
Initialize: $\hat{\theta}$, $\Delta^{(i)}$, and L
for time step $t = 1, 2, 3, ...$ **do**
Action: take an action based on
$\hat{p}(s'|s,a) \leftarrow \mathcal{N}(\hat{\theta}^T \Phi(s,a), \sigma^2 I)$
$\tilde{r}(s,a,s') \leftarrow R(s,a,s') + L\|\mathbf{I}_h(s,a,X_{t-1})\|$
Observation: Save the input-output pair $(s_{t+1}, \Phi_t(s_t, a_t))$
Estimate: Estimate $\hat{\theta}_{(i)}$, $\Delta^{(i)}$ (if not given), and L (if not given)

The pseudocode for this is shown in Algorithm 3.2.

3.4.2 Analysis

Following previous work [44, 27], we assume access to an exact planning algorithm. This assumption would be relaxed by using a planning method that provides an error bound. We assume that Algorithm 3.2 is used with least-squares estimation, which determines $\mathcal{L}$. We fix the distance function as $d(\widehat{P}(\cdot|s,a), P(\cdot|s,a)) = |E_{s' \sim \widehat{P}(\cdot|s,a)}[s'] - E_{s' \sim P(\cdot|s,a)}[s']|$ (since the unknown aspect is the mean, this choice makes sense). In the following, we use $\bar{n}$ to represent the average value of $\{n_{(1)}, ..., n_{(n_S)}\}$. The proofs are given in the appendix.

Lemma 3.3. (Sample complexity of PAC-MDP) For our problem setting, the PAC-MDP algorithm proposed by Strehl and Littman [44] and Li et al. [27] has sample complexity $\tilde{O}\left(\frac{n_S^2 \bar{n}^2}{\epsilon^5 (1-\gamma)^{10}}\right)$.

Theorem 3.2. (PAC-RMDP) Let $\mathcal{A}_t$ be the policy of Algorithm 3.2. Let

$$z = \max(h^2 \ln \frac{m^2 n_s h}{\delta}, \frac{L^2 n_S \bar{n} \ln^2 m}{\epsilon^3} \ln \frac{n_S}{\delta}).$$

Then, for all $\epsilon > 0$, for all $\delta = (0, 1)$, and for all $h \geq 0$,

1) for all but at most $m' = O\left(\frac{z L^2 n_S \bar{n} \ln^2 m}{\epsilon^3 (1-\gamma)^2} \ln^2 \frac{n_S}{\delta}\right)$ time steps (with $m \leq m'$), $V^{\mathcal{A}_t}(s_t) \geq V^{d*}_{\mathcal{L},t,h}(s_t) - \epsilon$, with probability at least $1 - \delta$, *and*

2) there exists $h^*(\epsilon, \delta) = O(\mathcal{P}(1/\epsilon, 1/\delta, 1/(1-\gamma), |\text{MDP}|))$ such that

$$|V^*(s_t) - V^{d*}_{\mathcal{L},t,h^*(\epsilon,\delta)}(s_t)| \leq \epsilon$$

with probability at least $1 - \delta$.

Corollary 3.3. (Anytime error bound) With probability at least $1-\delta$, if $h^2 \ln \frac{m^2 n_s h}{\delta} \leq \frac{L^2 n_S \bar{n} \ln^2 m}{\epsilon^3} \ln \frac{n_S}{\delta}$,

$$\epsilon_{t,h} = O\left(\sqrt[5]{\frac{L^4 n_S^2 \bar{n}^2 \ln^2 m}{t(1-\gamma)} \ln^3 \frac{n_S}{\delta}}\right);$$

otherwise,

$$\epsilon_{t,h} = O\left(\frac{h^2 L^2 n_S \bar{n} \ln^2 m}{t(1-\gamma)} \ln^2 \frac{n_S}{\delta}\right).$$

The anytime T-step average loss is equal to $\frac{1}{T}\sum_{t=1}^{T}(1-\gamma^{T+1-t})\epsilon_{t,h}$.

Corollary 3.4. (Explicit exploration runtime) With probability at least $1-\delta$, the explicit exploration runtime of Algorithm 3.2 is

$$O\left(\frac{h^2 L^2 n_S \bar{n} \ln m}{\epsilon^2 \Pr[A_k]} \ln^2 \frac{n_S}{\delta} \ln \frac{m^2 n_s h}{\delta}\right) = O\left(\frac{h^2 L^2 n_S \bar{n} \ln m}{\epsilon^3 (1-\gamma)} \ln^2 \frac{n_S}{\delta} \ln \frac{m^2 n_s h}{\delta}\right)$$

where A_K is the escape event defined in the proof of Theorem 3.2.

3.4.3 Experimental Examples

We consider two examples: the mountain car problem [49], which is a standard toy problem in the literature, and the HIV problem [12], which originates from a real-world problem. For both examples, we compare the proposed algorithm with a directly related PAC-MDP algorithm [44, 27]. For the PAC-MDP algorithm, we present the results with ϵ set to several theoretically meaningful values and one theoretically

non-meaningful value to illustrate its property[4]. We used $\delta = 0.9$ for the PAC-MDP and PAC-RMDP algorithms[5]. The ϵ-greedy algorithm is executed with $\epsilon = 0.1$. In the planning phase, L is estimated as $L \leftarrow \max_{s,s' \in \Omega} |\tilde{V}^{\mathcal{A}}(s) - \tilde{V}^{\mathcal{A}}(s')| / \|s - s'\|$, where Ω is the set of states that are visited in the planning phase (i.e., fitted value iteration and a greedy roll-out method). For both problems, more detailed descriptions of the experimental settings are available in the appendix.

Mountain Car

In the mountain car problem, the reward is negative everywhere except at the goal. To reach the goal, the agent must first travel far away, and must explore the world to learn this mechanism. Each episode consists of 2000 steps, and we conduct simulations for 100 episodes.

The numerical results are shown in Figure 3-2. As in the discrete case, we can see that the PAC-RMDP(h) algorithm worked well. The best performance, in terms of the total reward, was achieved by PAC-RMDP(10). Since this problem required a number of consecutive explorations, the random exploration employed by the ϵ-greedy algorithm did not allow the agent to reach the goal. As a result of exploration and the randomness in the environment, the PAC-MDP algorithm reached the goal several times, but kept exploring the environment to ensure near-optimality. From Figure 3-2, we can see that the PAC-MDP algorithm quickly converges to good behavior if we discard the theoretical guarantee (the difference between the values in the optimal value function had an upper bound of 120, and the total reward had an upper bound of 2000. Hence, $\epsilon > 2000$ does not yield a useful theoretical guarantee).

Simulated HIV Treatment

This problem is described by a set of six ordinary differential equations [12]. An action corresponds to whether the agent administers two treatments (RTIs and PIs) to patients (thus, there are four actions). Two types of exploration are required: one

[4]See footnote 3 on the consideration of different values of ϵ.

[5]We considered $\delta = [0.5, 0.8, 0.9, 0.95]$, but there was no change in any qualitative behavior of interest in our discussion.

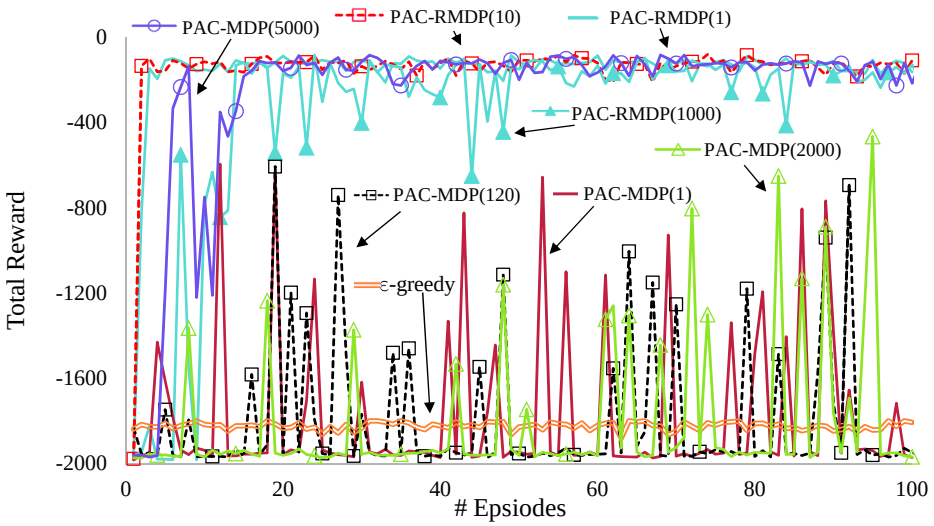


Figure 3-2: Total reward per episode for the mountain car problem with PAC-RMDP(h) and PAC-MDP(ϵ).

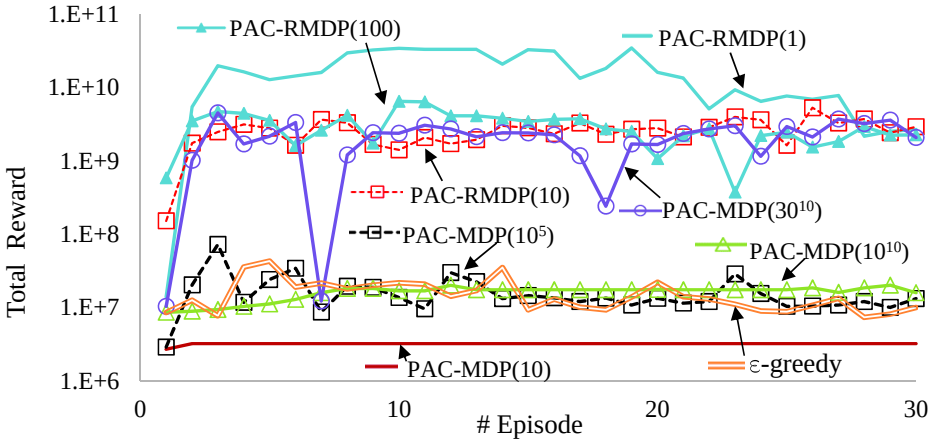


Figure 3-3: Total reward per episode for the HIV problem with PAC-RMDP(h) and PAC-MDP(ϵ).

to learn the effect of using treatments on viruses, and another to learn the effect of not using treatments on immune systems. Learning the former is necessary to reduce the population of viruses, but the latter is required to prevent the overuse of treatments, which weakens the immune system. Each episode consists of 1000 steps (i.e., days), and we conduct simulations for 30 episodes.

As shown in Figure 3-3, the PAC-MDP algorithm worked reasonably well with $\epsilon = 30^{10}$. However, the best total reward did not exceed 30^{10}, and so the PAC-MDP guarantee with $\epsilon = 30^{10}$ does not seem to be useful. The ϵ-greedy algorithm did not

work well, as this example required sequential exploration at certain periods to learn the effects of treatments.

Although the PAC-RMDP(h) algorithms worked well in our experimental examples with small h in this section and Section 3.3.3, it is possible to devise a problem in which the PAC-RMDP algorithm should be used with large h as illustrated in Appendix B.5. In extreme cases, the PAC-RMDP algorithms would reduce to PAC-MDP. Thus, the adjustable theoretical guarantee of PAC-RMDP(h) via the concept of reachability seems to be a reasonable objective.

Chapter 4

Conclusion

In this thesis, I have discussed two principles to tighten theoretical bounds, aiming to bridge the gap between theory and practice. For Bayesian optimization, I have taken advantage of *tighter* yet *unknown* bounds. As a result, I have presented the first GP-based optimization method with an exponential convergence rate $O\left(\lambda^{N+N_{gp}}\right)$ $(\lambda < 1)$ *without* the need of auxiliary optimization and the δ-cover sampling. For exploration in MDPs, I have introduced the concept of *reachability in model learning* and used this concept for re-considering the optimality criteria. As a result, I have proposed the PAC–RMDP framework, and two algorithms for both discrete and continuous state representations that improve the previous theoretical bounds and empirical performance.

The two principles used in this thesis (as illustrated in Figure 1-1) are different and not intended to be exclusive. That is, one may combine these two approaches to improve a bound-based method. Although each principle is discussed in the consideration of Bayesian optimization or exploration in MDP, it would be interesting to see if similar approaches are applicable to other types of bound-based methods such as planning algorithms (e.g., A* search and the UCT or FSSS algorithm [50]).

In Chapter 2, from the viewpoint of the global optimization community, I have provided a practically oriented analysis framework, enabling us to see why *not* relying on a particular bound is advantageous, and how a non-tight bound can still be useful (in Remarks 2.1, 2.2 and 2.3). In Chapter 3, from the viewpoint of optimal

exploration, I have provided a theoretical framework that is applicable to more practical and complex models than the existing framework. Whereas the development of algorithms with traditional objectives (PAC-MDP or regret bounds) requires the consideration of confidence intervals, the proposed objective, PAC-RMDP(h), concerns a set of h-reachable models. For a flexible structured model, the derivation of the confidence interval would be a difficult task, but a set of h-reachable models can simply be computed (or approximated) via lookahead by using the model update rule. Thus, future work includes the derivation of a PAC-RMDP algorithm with a more structured model.

Appendix A

Appendix – Bayesian Optimization

In this appendix, we provide the proofs of the theoretical results presented in Chapter 2 – Bayesian Optimization with Exponential Convergence. Along the way, we also prove regret bounds for a general class of algorithms, the result of which may be used to design a new algorithm.

We first provide a known property of the upper confidence bound of GP.

Lemma 2.1. (Bound Estimated by GP) According to the belief encoded in the GP prior/posterior[1], for any x, $f(x) \leq \mathcal{U}(x|\mathcal{D})$ holds during the execution of Algorithm 2.1 with probability at least $1 - \eta$.

Proof. It follows the proof of lemma 5.1 of [41]. From the property of the standard gaussian distribution, $\Pr(f(x) > \mathcal{U}(x|\mathcal{D})) < \frac{1}{2}e^{-\varsigma_M^2/2}$. Taking union bound on the entire execution of Algorithm 2.1, $\Pr(f(x) > \mathcal{U}(x|\mathcal{D}) \ \forall M \geq 1) < \frac{1}{2}\sum_{M=1}^{\infty} e^{-\varsigma_M^2/2}$. Substituting $\varsigma_M = \sqrt{2\log(\pi^2 M^2/12\eta)}$, we obtain the statement. □

Our algorithm has a concrete division procedure in line 27 of Algorithm 2.1. However, one may improve the algorithm with different division procedures. Accordingly, we first derive abstract version of regret bound for the IMGPO (Algorithm 2.1) under a family of division procedures that satisfy Assumptions 2.3 and 2.4. After that, we provide a proof for the main results in the thesis.

[1]Thus, the probability in this analysis should be seen as that of *the subjective view*. If we assume that f is indeed a sample from the GP, we have the same result with *the objective view* of probability.

A.1 Proofs for Family of Division Procedures

In this section, we modify the result obtained by [31]. Let $x_{h,i}$ to be any point in the region covered by the i^{th} hyperinterval at depth h, and $x^*_{h,i}$ be the global optimizer that may exist in the i^{th} hyperinterval at depth h. The previous work provided the regret bound of the SOO algorithm with a family of division procedure that satisfies the following two assumptions.

Assumption 2.3. (Decreasing diameter) There exists a diameter function $\delta(h) > 0$ such that, for any hyperinterval $\omega_{h,i} \subset \Omega$ and its center $c_{h,i} \in \omega_{h,i}$ and any $x_{h,i} \in \omega_{h,i}$, we have $\delta(h) \geq \sup_{x_{h,i}} \ell(x_{h,i}, c_{h,i})$ and $\delta(h-1) \geq \delta(h)$ for all $h \geq 1$.

Assumption 2.4. (Well-shaped cell) There exists $\nu > 0$ such that any hyperinterval $\omega_{h,i}$ contains at least an ℓ-ball of radius $\nu\delta(h)$ centered in $\omega_{h,i}$.

Thus, in this section, hyperinterval is not restricted to hyperrectangle. We now revisit the definitions of several terms and variables used in [31]. Let the ϵ-optimal space X_ϵ be defined as $X_\epsilon := \{x \in \Omega : f(x) + \epsilon \geq f(x^*)\}$. That is, the ϵ-optimal space is the set of input vectors whose function value is at least ϵ-close to the global optima. To bound the number of hyperintervals relevant to this ϵ-optimal space, we define a near-optimality dimension as follows.

Definition 2.3. (Near-optimality dimension) The near-optimality dimension is the smallest $d > 0$ such that, there exists $C > 0$, for all $\epsilon > 0$, the maximum number of disjoint ℓ-balls of radius $\nu\epsilon$ with center in the ϵ-optimal space X_ϵ is less than $C\epsilon^{-d}$.

3

Finally, we define the set of δ-optimal hyperintervals $I_{\delta(h)}$ as $I_{\delta(h)} := \{\omega_{h,i} \ni c_{h,i} : f(c_{h,i}) + \delta(h) \geq f(x^*)\}$. The δ-optimal hyperinterval $I_{\delta(h)}$ is used to relate the hyperintervals to the ϵ-optimal space. Indeed, the δ-optimal hyperinterval $I_{\delta(h)}$ is almost identical to the $\delta(h)$-optimal space $X_{\delta(h)}$, except that $I_{\delta(h)}$ is focused on the center points whereas $X_{\delta(h)}$ considers the whole input vector space. In the following,

we use $|I_{\delta(h)}|$ to denote the number of $I_{\delta(h)}$ and derive its upper bound.

Lemma 2.2. (Lemma 3.1 in [31]) Let d be the near-optimality dimension and C denote the corresponding constant in Definition 2.1. Then, the number of δ-optimal hyperintervals is bounded by $|I_{\delta(h)}| \leq C\delta(h)^{-d}$.

We are now ready to present the main result in this section. In the following, we use the term *optimal hyperinterval* to indicate a hyperinterval that contains a global optimizer x^*. We say a hyperinterval is *dominated* by other intervals when it is rejected or not selected in step (i)-(iii). In Lemma 2.3, we bound the maximum size of the optimal hyperinterval. From Assumption 2.1, this can be translated to the regret bound, as we shall see in Theorem 2.2.

Lemma 2.3. Let $\Xi_n \leq \min(\Xi, \Xi_{max})$ be the largest ξ used so far with n total node expansions. Let h_n^* be the depth of the deepest expanded node that contains a global optimizer x^* after n total node expansions (i.e., $h_n^* \leq n$ determines the size of the *optimal hyperinterval*). Then, with probability at least $1-\eta$, h_n^* is bounded below by some h' that satisfies

$$n \geq \sum_{\tau=1}^{\sum_{l=0}^{h'+\Xi}|I_l|} \rho_\tau.$$

Proof. Let T_h denote the time at which the optimal hyperinterval is further divided. We prove the statement by showing that the time difference $T_{h+1} - T_h$ is bounded by the number of δ-optimal hyperintervals. To do so, we first note that there are three types of hyperinterval that can dominate an optimal hyperinterval $c_{h+1,*}$ during the time $[T_h, T_{h+1}-1]$, all of which belong to δ-optimal hyperintervals I_δ. The first type has the same size (i.e., same depth h), $c_{h+1,i}$. In this case,

$$f(c_{h+1,i}) \geq f(c_{h+1,*}) \geq f(x^*_{h+1,*}) - \delta(h+1),$$

where the first inequality is due to line 10 (step (i)) and the second follows Assumptions 2.1 and 2.2. Thus, it must be $c_{h+1,i} \in I_{h+1}$. The second case is where the optimal hyperinterval may be dominated by a hyperinterval of larger size (depth $l < h+1$),

$c_{l,i}$. In this case, similarly,

$$f(c_{l,i}) \geq f(c_{h+1,*}) \geq f(x^*_{h+1,*}) - \delta(l),$$

where the first inequality is due to lines 11 to 12 (step (ii)) and thus $c_{l,i} \in I_l$. In the final scenario, the optimal hyperinterval is dominated by a hyperinterval of smaller size (depth $h+1+\xi$), $c_{h+1+\xi,i}$. In this case,

$$f(c_{h+1+\xi,i}) \geq z(h+1,*) \geq f(x^*_{h+1,*}) - \delta(h+1+\xi)$$

with probability at least $1-\eta$ where $z(\cdot,\cdot)$ is defined in line 21 of Algorithm 2.1. The first inequality is due to lines 19 to 23 (step (iii)) and the second inequality follows Lemma 2.1 and Assumptions 2.1 and 2.3. Hence, we can see that $c_{h+1+\xi,i} \in I_{h+1+\xi}$.

For all of the above arguments, the temporarily assigned $\mathcal{U}$ under GP has no effect. This is because the algorithm still covers the above three types of δ-optimal hyperintervals I_δ, as $\mathcal{U} \geq f$ with probability at least $1-\eta$ (Lemma 2.1). However, these are only expanded based on f because of the temporary nature of $\mathcal{U}$. Putting these results together,

$$T_{h+1} - T_h \leq \sum_{\tau=1}^{\sum_{l=1}^{h+1+\Xi_n} |I_{\delta(l)}|} \rho_\tau.$$

Since if one of the I_δ is divided during $[T_h, T_{h+1}-1]$, it cannot be divided again during another time period,

$$\sum_{h=0}^{h^*_n} T_{h+1} - T_h \leq \sum_{\tau=1}^{\sum_{l=1}^{h^*_n+1+\Xi_n} |I_l|} \rho_\tau,$$

where on the right-hand side, the summation $\sum_{h=0}^{h^*_n}$ was combined into another $\sum_{\tau=1}^{\sum_{l=1}^{h+1+\Xi_n} |I_{\delta(l)}|}$, because each h in the summation refers to the same δ-optimal interval $I_{\delta(l)}$ with $l \leq h^*_n + 1 + \Xi_n$, and should not be double-counted. As $\sum_{h=0}^{h^*_n} T_{h+1} - T_h = T_{h^*_n+1} - T_0$, $T_0 = 1$ and $|I_{\delta(0)}| = 1$,

$$T_{h^*_n+1} \leq 1 + \sum_{\tau=1}^{\sum_{l=1}^{h^*_n+1+\Xi_n} |I_l|} \rho_\tau \leq \sum_{\tau=1}^{\sum_{l=0}^{h^*_n+1+\Xi_n} |I_l|} \rho_\tau.$$

As $T_{h_n^*+1} > n$ by definition, for any h' such that $\sum_{\tau=1}^{\sum_{l=0}^{h'+\Xi_n}|I_l|}\rho_\tau \leq n < \sum_{\tau=1}^{\sum_{l=0}^{h_n^*+1+\Xi_n}|I_l|}\rho_\tau$, we have $h_n^* > h'$. □

With Lemmas 2.2 and 2.3, we are ready to present a finite regret bound with the family of division procedures.

Theorem 2.2. Assume Assumptions 2.1, 2.3, and 2.4. Let $h(n)$ be the smallest integer h such that

$$n \leq \sum_{\tau=1}^{C\sum_{l=0}^{h+\Xi_n}\delta(l)^{-d}} \rho_\tau.$$

Then, with probability at least $1-\eta$, the regret of the IMGPO with any general division procedure is bounded as

$$r_n \leq \delta(h(n)-1).$$

Proof. Let $c(n)$ and $c_{h_n^*,*}$ be the center point expanded at the nth expansion and the optimal hyperinterval containing a global optimizer x^*, respectively. Then, from Assumptions 2.1, 2.3, and 2.4, $f(c(n)) \geq f(c_{h_n^*,*}) \geq f^* - \delta(h_n^*)$, where f^* is the global optima. Hence, the regret bound is $r_h \leq \delta(h_n^*)$. To find a lower bound for the quantity h_n^*, we first relate $h(n)$ to Lemma 2.3 by

$$n > \sum_{\tau=1}^{C\sum_{l=0}^{h(n)+\Xi_n-1}\delta(l)^{-d}} \rho_\tau \geq \sum_{\tau=1}^{\sum_{l=0}^{h(n)+\Xi_n-1}|I_l|} \rho_\tau,$$

where the first inequality comes from the definition of $h(n)$, and the second follows from Lemma 2.2. Then, from Lemma 2.3, we have $h_n^* \geq h(n)-1$. Therefore, $r_n \leq \delta(h_n^*) \leq \delta(h(n)-1)$. □

Assumption 2.5. (Decreasing diameter revisit) The decreasing diameter defined in Assumption 2.3 can be written as $\delta(h) = c_1\gamma^{h/D}$ for some $c_1 > 0$ and $\gamma < 1$ with a division procedure that requires c_2 function evaluations per node expansion.

Corollary 2.1. Assume Assumptions 2.1, 2.3, 2.4, and 2.5. Then, if $d = 0$, with probability at least $1 - \eta$,

$$r_N \leq O\left(\exp\left(-\frac{N + N_{gp}}{c_2 C D \bar{\rho}_t}\right)\right).$$

If $d > 0$, with probability at least $1 - \eta$,

$$r_N \leq O\left(\left(\frac{1}{N + N_{gp}}\right)^{1/d}\left(-\frac{c_2 C \bar{\rho}_t}{1 - \gamma^{d/D}}\right)^{1/d}\gamma^{-\frac{1}{D}}\right).$$

Proof. For the case $d = 0$, we have $n \leq \sum_{\tau=1}^{C\sum_{l=0}^{h(n)+\Xi_n}\delta(l)^{-d}} \rho_\tau \leq \sum_{\tau=1}^{C(h(n)+\Xi_n+1)} \bar{\rho}_t$, where the first inequality follows from the definition of $h(n)$, and the second comes from the definition of $\bar{\rho}_t$ and the assumption $d = 0$. The second inequality holds for $\bar{\rho}_t$ that only considers ρ_τ with $\tau \leq t$. This is computable, because $\tau \leq t$ by construction. Indeed, the condition of Lemma 2.3 implies $t \geq \sum_{l=0}^{h'+\Xi_n} |I_l|$. Therefore, the two inequalities hold, and we can deduce that $h(n) \geq \frac{n}{C\bar{\rho}_t} - \Xi_n - 1$ by algebraic manipulation. By Assumption 2.5, $n = (N + N_{gp})/c_2$. With this, substituting the lower bound of $h(n)$ into the statement of Theorem 2.2 with Assumption 2.5,

$$r_N \leq c_1 \exp\left(-\left[\frac{N + N_{gp}}{c_2 D}\frac{1}{C\bar{\rho}_t} - \Xi_n - 2\right]\ln\frac{1}{\gamma}\right).$$

Similarly, for the case $d > 0$,

$$n \leq \sum_{\tau=1}^{C\sum_{l=0}^{h(n)+\Xi_n}\delta(l)^{-d}} \rho_\tau \leq \sum_{\tau=1}^{c^{-d}C\frac{\gamma^{-(h(n)+\Xi_n+1)d/D}-1}{\gamma^{-d/D}-1}} \bar{\rho}_t,$$

and hence $c\gamma^{\frac{h(n)+\Xi_n}{D}} \leq \left(\frac{n(1-\gamma^{d/D})}{C\bar{\rho}_t}\right)^{-1/d}$ by algebraic manipulation. Substituting this into the result of Theorem 2.2, we arrive at the desired result. □

A.2 Proofs for a Concrete Division Procedure

In this section, we prove the main result in the thesis. In Theorem 2.1, we show that the exponential convergence rate bound $O\left(\lambda^{N+N_{gp}}\right)$ with $\lambda < 1$ is achieved *without* Assumptions 2.3, 2.4 and 2.5 and *without* the assumption that $d = 0$.

Theorem 2.1. Assume Assumptions 2.1 and 2.2. Let $\beta = \sup_{x,x' \in \Omega} \frac{1}{2}\|x - x'\|_\infty$. Let $\lambda = 3^{-\frac{\alpha}{2C\bar{\rho}_t D}} < 1$. Then, *without* Assumptions 2.3, 2.4 and 2.5 and *without* the assumption on d, with probability at least $1 - \eta$, the regret of IMGPO with the division procedure in Algorithm 2.1 is bounded as

$$r_N \leq L(3\beta D^{1/p})^\alpha \exp\left(-\alpha\left[\frac{N + N_{gp}}{2C\bar{\rho}_t D} - \Xi_n - 2\right]\ln 3\right) = O\left(\lambda^{N+N_{gp}}\right).$$

Proof. To prove the statement, we show that Assumptions 3, 4, and 5 can all be satisfied while maintaining $d = 0$.

From Assumption 2 (i), and based on the division procedure that the algorithm uses,

$$\sup_{x \in \omega_{h,i}} \ell(x, c_{h,i}) \leq \sup_{x \in \omega_{h,i}} L||x - c_{h,i}||_p^\alpha \leq L\left(3^{-\lfloor h/D \rfloor}\beta D^{1/p}\right)^\alpha.$$

This upper bound corresponds to the diagonal length of each hyperrectangle with respect to p-norm, where $3^{-\lfloor h/D \rfloor}\beta$ corresponds to the length of the longest side. We fix the form of δ as $\delta(h) = L3^\alpha D^{\alpha/p} 3^{-h\alpha/D}\beta^\alpha \geq L(3^{-\lfloor h/D \rfloor}\beta D^{1/p})^\alpha$, which satisfies Assumption 3.

This form of $\delta(h)$ also satisfies Assumption 5 with $\gamma = 3^{-\alpha}$ and $c_1 = L3^\alpha D^{\alpha/p}\beta^\alpha$.

Every hyperrectangle contains at least one ℓ-ball with a radius corresponding to the length of the shortest side of the hyperrectangle. Thus, we have at least one ℓ-ball of radius $\nu\delta(h) = L3^{-\alpha\lceil h/D \rceil} \geq L3^{-\alpha}3^{-\alpha h/D}$ for every hyperrectangle with $\nu \geq 3^{-2\alpha}D^{-\alpha/p}$. This satisfies Assumption 4.

Finally, we show that $d = 0$. The set of δ-optimal hyperintervals $I_{\delta(h)}$ is contained by

the $\delta(h)$-optimal space $X_{\delta(h)}$ as

$$I_{\delta(h)} = \{c \in \Omega : f(x^*) - f(c) \leq \delta(h), c \text{ is the center point of the interval}, \omega_{h,i}, \text{for some } (h,i)\}$$
$$\subseteq \{x \in \Omega : f(x^*) - f(x) \leq \delta(h)\} = X_{\delta(h)}$$

Let θ be a value that satisfies Assumption 2 (ii) (which is nonzero). Consider an ℓ-ball of radius $\frac{\delta(h)}{\theta}$ at x^*, which is a set $\{x \in \Omega \mid \theta\ell(x, x^*) \leq \delta(h)\}$. Since $\theta\ell(x, x^*) \leq f(x^*) - f(x)$ by Assumption 2 (ii), the $\delta(h)$-optimal space $X_{\delta(h)}$ is covered by an ℓ-ball of radius $\frac{\delta(h)}{\theta}$. Therefore, $I_{\delta(h)} \subseteq X_{\delta(h)} \subseteq$ (an ℓ-ball of radius $\frac{\delta(h)}{\theta}$ at x^*). By Assumption 2 (i), the volume V of an ℓ-ball of radius $\nu\delta(h)$ is proportional to $(\nu\delta(h))^D$ as $V_D^p(\nu\delta(h)) = (2\nu\delta(h)\Gamma(1+1/p))^D/\Gamma(1+D/p)$. Thus, the number of disjoint ℓ-balls of radius $\nu\delta(h)$ that fit in $X_{\delta(h)}$ is at most $\lceil(\frac{\delta(h)}{\theta\nu\delta(h)})^D\rceil = \lceil(\theta\nu)^{-D}\rceil$. Therefore, the number of ℓ-balls does not depend on $\delta(h)$ in this case, which means $d = 0$.

Now that we have satisfied Assumptions 3, 4, and 5 with $d = 0$, $\gamma = 3^{-\alpha}$, and $c_1 = L3^\alpha D^{\alpha/p}\beta^\alpha$, we follow the proof of Corollary 1 and deduce the desired statement. □

Appendix B

Appendix – Exploration in MDP

In this appendix, we provide the proofs of the theoretical results, additional experimental results and discussions that were deferred in Chapter 3 – Bounded Optimal Exploration in MDP.

B.1 Proofs of Propositions 1 and 2

In this section, we present the proofs of Propositions 1 and 2.

Proposition 3.1. (PAC-MDP) PAC-RMDP($h^*(\epsilon, \delta)$) implies PAC-MDP, where $h^*(\epsilon, \delta)$ is given in Definition 3.1.

Proof. For any PAC-RMDP($h^*(\epsilon, \delta)$) algorithm, Definition 3.1 implies

$$V^{\mathcal{A}}(s_t) \geq V^*_{h^*}(s_t) - \epsilon \geq V^*(s_t) - 2\epsilon$$

with probability at least $1 - 2\delta$ for all but polynomial time steps. This satisfies the condition of the PAC-MDP. □

Proposition 3.2. (Near-Bayes optimality) Consider the model-based Bayesian reinforcement learning [46]. Let H be a planning horizon in the belief space b. Assume that the Bayesian optimal value function, $V^*_{b,H}$, converges to the H-reachable optimal

function such that, for all $\epsilon > 0$, $|V^*_{\mathcal{L},t,H}(s_t) - V^*_{b,H}(s_t, b_t)| \leq \epsilon$ for all but polynomial time steps. Then, a PAC-RMDP(H) algorithm with a policy $\mathcal{A}_t$ obtains an expected cumulative reward $V^{\mathcal{A}_t}(s_t) \geq V^*_{b,H}(s_t, b_t) - 2\epsilon$ for all but polynomial time steps with probability at least $1 - \delta$.

Proof. It directly follows Definition 3.1 and the assumption. For all but polynomial time steps, with probability at least $1 - \delta$, $V^{\mathcal{A}}(s_t) \geq V^*_{\mathcal{L},t,H}(s_t) - \epsilon \geq V^*_{b,H}(s_t, b_t) - 2\epsilon$. □

B.2 Relationship to Bounded Rationality and Bounded Optimality

As the concept of PAC-RMDP considers the inherent limitations of a decision maker, it shares properties with the concepts of bounded rationality [38] and bounded optimality [36].

Bounded rationality and bounded optimality focus on limitations in the planning phase (e.g., computational resources). In contrast, PAC-RMDP considers limitations in the learning phase (e.g., the agent's lifetime). As in the case of bounded rationality, the performance guarantee of a PAC-RMDP(h) algorithm can be arbitrary, depending on the choice of h. On the contrary, bounded optimality solves the problem of arbitrariness, seemingly at the cost of applicability. It requires a strong notion of optimality, similar to instance optimality; roughly, we must find the *optimal algorithm* given the available computational resources. Automated optimization over the set of algorithms is a difficult task. Zilberstein [54] claims that bounded optimality is difficult to achieve, resulting in very few successful examples, and is not, in practice, as promising as other bounded rational methods. However, in future research, it would be interesting to compare PAC-RMDP with a possible relaxation of PAC-MDP based on a concept similar to bounded optimality.

B.3 Corresponding Notions of Regret and Average Loss

In the definition of PAC-RMDP(h), our focus is on *learning* useful models, enabling us to obtain high rewards in a short period of time. Instead, one may wish to guarantee the worst total reward *in a given time horizon* T. There are several ways to achieve this goal. One solution is to minimize the expected T-step *regret bound* $r^{\mathcal{A}}(T)$, given by

$$r^{\mathcal{A}}(T) \geq V^*(s_0, T) - V^{\mathcal{A}}(s_0, T). \tag{B.1}$$

In this case, $V^*(s_0, T) = E[\sum_{i=0}^{T} \gamma^i R\left(s_i^*, \pi^*(s_i), s_{i+1}^*\right)]$, where the sequence of states $s_0^*, s_1^*, ..., s_T^*$ with $s_0^* = s_0$ is generated when the agent follows the optimal policy π^* from s_0, and $V^{\mathcal{A}}(s_0, T) = E[\sum_{i=0}^{T} \gamma^i R\left(s_i, \mathcal{A}_i(s_i), s_{i+1}\right)]$, where the sequence of states $s_0, s_1, ..., s_T$ is generated when the agent follows policy $\mathcal{A}_i$. Since one mistake in the early stages may make it impossible to return to the optimal state sequence s_i^*, all the regret approaches in the literature rely on some reachability assumptions in the state space; for example, Jaksch et al. [15] assumed that every state was reachable from every other state within a certain (average) number of steps.

Another approach is to minimize the expected T-step *average loss bound* $r^{\mathcal{A}}(T)$, which obviates the need for any reachability assumptions in the state space:

$$\ell^{\mathcal{A}}(T) \geq \frac{1}{T} \sum_{t=0}^{T} \left[V^*(s_t, T) - V^{\mathcal{A}}(s_t, T)\right], \tag{B.2}$$

where s_t is the state visited by algorithm $\mathcal{A}$ at time t. The value functions inside the sum are defined as $V^*(s_t, T) = E[\sum_{i=0}^{T-t} \gamma^i R\left(s_{t+i}^*, \pi^*(s_{t+i}), s_{t+i+1}^*\right)]$ with $s_t^* = s_t$ and $V^{\mathcal{A}}(s_0, T) = E[\sum_{i=0}^{T-t} \gamma^i R\left(s_{t+i}^*, \mathcal{A}_t(s_{t+i}), s_{t+i+1}\right)]$. By averaging the T-step regrets (i.e., losses) of the T initial states $s_0, s_1, ..., s_T$ visited by $\mathcal{A}$, the average loss mitigates the effects of irreversible mistakes in the early stages that may dominate the regret.

The expected h-reachable regret bound $r_h^{\mathcal{A}}(T)$ and average loss bound $\ell_h^{\mathcal{A}}(T)$ are

defined as

$$r_h^{\mathcal{A}}(T) \geq V^*_{\mathcal{L},t,h}(s_0, T) - V^{\mathcal{A}}(s_0, T)$$

and $\ell_h^{\mathcal{A}}(T) \geq \frac{1}{T}\sum_{t=1}^{T}\left[V^*_{\mathcal{L},t,h}(s_t, T) - V^{\mathcal{A}}(s_t, T)\right]$. That is, they are the same as the standard expected regret and average loss, respectively, with the exception that the optimal value function V^* has been replaced by the h-reachable optimal value function $V^*_{\mathcal{L},t,h}(s_t)$.

While the definition of PAC-RMDP(h) focuses on exploration, the proposed PAC-RMDP(h) algorithms maintain anytime expected h-reachable average loss bounds and anytime error bounds, and thus the performances of our algorithms are expected to improve with time, rather than after some number of exploration steps.

B.4 Proofs of Theoretical Results for Algorithm 3.1

We first verify the main properties of Algorithm 3.1 and then analyze a practically relevant property of the algorithm in the subsection of Further Discussion. We assume that Algorithm 3.1 is used with the sample mean estimator, which determines $\mathcal{L}$.

Main Properties

To compare the results with those of past studies, we assume that $R_{max} \leq c$ for some fixed constant c. The effect of this assumption can be seen in the proof of Theorem 3.1. Algorithm 3.1 requires no input parameter related to ϵ and δ. This is because the required degree of optimism can be determined independently of the unknown aspect of the world. This means that Theorem 3.1 holds at any time during an execution for a pair of corresponding ϵ and δ.

Lemma 3.1. (Optimism) For all $s \in S$ and for all $t, h \geq 0$, the internal value $\tilde{V}^{\mathcal{A}_t}(s)$ used by Algorithm 3.1 is at least the h-reachable optimal value $V^*_{\mathcal{L},t,h}(s)$; $\tilde{V}^{\mathcal{A}_t}(s) \geq V^*_{\mathcal{L},t,h}(s)$.

Proof. The claim follows directly from the construction of Algorithm 3.1. It can

be verified by induction on each step of the value iteration or the roll-out in a planning algorithm. □

Theorem 3.1. (PAC-RMDP) Let $\mathcal{A}_t$ be a policy of Algorithm 3.1. Let

$$z = \max(h, \frac{\ln(2^{|S|}|S||A|/\delta)}{\epsilon(1-\gamma)}).$$

Then, for all $\epsilon > 0$, for all $\delta = (0, 1)$, and for all $h \geq 0$,

1) for all but at most $O\left(\frac{z|S||A|}{\epsilon^2(1-\gamma)^2} \ln \frac{|S||A|}{\delta}\right)$ time steps, $V^{\mathcal{A}_t}(s_t) \geq V^*_{\mathcal{L},t,h}(s_t) - \epsilon$, with probability at least $1 - \delta$, *and*

2) there exists $h^*(\epsilon, \delta) = O(\mathcal{P}(1/\epsilon, 1/\delta, 1/(1-\gamma), |\text{MDP}|))$ such that

$$|V^*(s_t) - V^*_{\mathcal{L},t,h^*(\epsilon,\delta)}(s_t)| \leq \epsilon$$

with probability at least $1 - \delta$.

Proof. Let K be a set of state-action pairs where the agent has at least m samples (this corresponds to *the set of known state-action pairs* described by Kearns and Singh [22]). With the boundary condition $\overline{V^{\mathcal{A}}}(s, 0) = 0$, define the mixed value function $\overline{V^{\mathcal{A}}}(s, H)$ with a finite horizon $H' = \frac{1}{1-\gamma} \ln \frac{6R_{max}}{\epsilon(1-\gamma)}$ as

$$\overline{V^{\mathcal{A}}}(s, H') = \begin{cases} \sum_{s'} P(s'|s, \mathcal{A}(s))[R(s, \mathcal{A}(s), s') + \gamma \overline{V^{\mathcal{A}}}(s', H' - 1)] & \text{if } (s, \mathcal{A}(s)) \in K \\ \max_{\tilde{P} \in \mathcal{M}_{\mathcal{L},t,h,(s,a)}} \sum_{s'} \tilde{P}(s'|s, \mathcal{A}(s))[R(s, \mathcal{A}(s), s') + \gamma \overline{V^{\mathcal{A}}}(s', H' - 1)] & \text{otherwise} \end{cases}$$

Let A_K be the escape event in which a pair $(s, a) \notin K$ is generated for the first time when starting at state s_t, following policy $\mathcal{A}_t$, and transitioning based on the true

dynamics P for H' steps. Then, for all $t, h \geq 0$, with probability at least $1 - \delta/2$,

$$\begin{aligned} V^{\mathcal{A}_t}(s_t) &\geq \overline{V^{\mathcal{A}_t}}(s_t, H') - \frac{R_{max}}{1-\gamma} \Pr(A_k) - \frac{\epsilon}{6} \\ &\geq \tilde{V}^{\mathcal{A}_t}(s_t) - \frac{R_{max}}{1-\gamma} \Pr(A_k) - \frac{\epsilon}{3} - \frac{R_{max}}{1-\gamma} \left(\frac{h}{m} + \sqrt{\frac{2 \ln(2^{|S|+1}|S||A|/\delta)}{m}} \right) \\ &\geq V^*_{\mathcal{L},t,h}(s_t) - \frac{R_{max}}{1-\gamma} \Pr(A_k) - \frac{\epsilon}{3} - \frac{R_{max}}{1-\gamma} \left(\frac{h}{m} + \sqrt{\frac{2 \ln(2^{|S|+1}|S||A|/\delta)}{m}} \right). \end{aligned}$$

The first inequality follows from the fact that $V^{\mathcal{A}_t}(s_t)$ and $\overline{V}^{\mathcal{A}_t}(s_t)$ are only different when event A_K occurs, and their difference is bounded above by $\frac{R_{max}}{1-\gamma}$ (this is the upper bound on the value $\tilde{V}(s_t)$). Furthermore, the finite horizon approximation adds an error of $1/6\epsilon$. A more detailed argument only involves algebraic manipulations that mirror the proofs given by Strehl and Littman (2008, Lemma 3) and Kearns and Singh (2002, Lemma 2).

The second inequality follows from the fact that $\overline{V}^{\mathcal{A}}$ is different from $\tilde{V}^{\mathcal{A}}$ only for the state-action pairs $(s, a) \in K$, for which $\tilde{V}^{\mathcal{A}_t}(s_t)$ deviates from $\overline{V}^{\mathcal{A}_t}(s_t)$ by at most $\frac{R_{max}}{1-\gamma}(\frac{h}{m} + \sqrt{2 \ln(2^{|S|+1}|S||A|/\delta)/m})$ with probability at least $1 - \delta/2$. This is because $|\tilde{V}^{\mathcal{A}_t}(s_t) - V^{\mathcal{A}_t}_{\mathcal{L},t,0}(s_t)| \leq \frac{R_{max}}{1-\gamma}\frac{h}{m}$ with certainty, and $|V^{\mathcal{A}_t}_{\mathcal{L},t,0}(s_t) - V^{\mathcal{A}_t}(s_t)| \leq \frac{R_{max}}{1-\gamma}\sqrt{2 \ln(2^{|S|+1}|S||A|/\delta)/m}$ with probability at least $1 - \delta/2$ (the later is due to the result of Weissman et al. (2003, Theorem 2.1) and the union bound for state-action pairs).

The third inequality follows from Lemma 3.1.

Therefore, if $h \leq \sqrt{2m \ln(2^{|S|+1}|S||A|/\delta)}$,

$$V^{\mathcal{A}_t}(s_t) \geq V^*_{\mathcal{L},t,h}(s_t) - \frac{R_{max}}{1-\gamma} \Pr(A_k) - \frac{\epsilon}{3} - \frac{2R_{max}}{1-\gamma} \sqrt{\frac{2 \ln(2^{|S|+1}|S||A|/\delta)}{m}}.$$

If $h > \sqrt{2m \ln(2^{|S|+1}|S||A|/\delta)}$,

$$V^{\mathcal{A}_t}(s_t) \geq V^*_{\mathcal{L},t,h}(s_t) - \frac{R_{max}}{1-\gamma} \Pr(A_k) - \frac{\epsilon}{3} - \frac{2R_{max}}{1-\gamma} \frac{h}{m}.$$

Let us consider the case where $h \leq \sqrt{2m \ln(2^{|S|+1}|S||A|/\delta)}$. We fix m as

$$m = \frac{72 R_{max}^2 \ln(2^{|S|+1}|S||A|/\delta)}{\epsilon^2 (1-\gamma)^2}$$

to give $\frac{\epsilon}{3}$ in the last term on the right-hand side. If $\Pr(A_K) \leq \frac{\epsilon(1-\gamma)}{3R_{max}}$ for all t, $V^{\mathcal{A}_t}(s_t) \geq V^*_{\mathcal{L},t,h}(s_t) - \epsilon$ with probability at least $1-\delta/2$. For the case where $\Pr(A_K) > \frac{\epsilon(1-\gamma)}{3R_{max}}$ for some t, we define an independent random event A'_K such that $\Pr(A'_K) = \frac{\epsilon(1-\gamma)}{3R_{max}} < \Pr(A_K)$. According to the Chernoff bound, for all $k \geq 4$, with probability at least $1-\delta/2$, the event A_K will occur at least k times after $\frac{2k}{\Pr(A'_K)} \ln \frac{2}{\delta}$ time steps. Thus, by applying the union bound on $|S|$ and $|A|$, we have a probability of at least $1-\delta/2$ of event A_K occurring at least m times for all state-action pairs after $O\left(\frac{m|S||A|}{\Pr(A'_k)} \ln \frac{|S||A|}{\delta}\right) = O\left(\frac{mR_{max}|S||A|}{\epsilon(1-\gamma)} \ln \frac{|S||A|}{\delta}\right)$ time steps.

Let us carefully consider what this means. Whenever A_K occurs, the sample is used to minimize the error between $V^{\mathcal{A}}$ and $\tilde{V}^{\mathcal{A}}$ by the definition of A_K. Since $\tilde{V}(s) \geq V^*_{\mathcal{L},t,h}(s)$ holds at any time, whenever A_K occurs, the sample is used to reduce the error in $V^{\mathcal{A}_t}(s_t) \geq \tilde{V}^{\mathcal{A}_t}(s_t) - (\text{error}) \geq V^*_{\mathcal{L},t,h}(s_t) - (\text{error})$ (note that if $\tilde{V}(s) \geq V^*_{\mathcal{L},t,h}(s)$ holds randomly, this event must occur concurrently with A_K to reduce the error on the right-hand side). Thus, after this number of time steps, $Pr(A_K)$ goes to zero with probability at least $1-\delta/2$. Hence, from the union bound, the above inequality becomes $V^{\mathcal{A}}(s_t) \geq V^*_{\mathcal{L},t,h}(s_t) - \frac{2}{3}\epsilon$ with probability at least $1-\delta$.

For the case where $h > \sqrt{2m \ln(2^{|S|+1}|S||A|/\delta)}$, we fix $m = \frac{hR_{max}}{6\epsilon(1-\gamma)}$. The rest of the proof follows that for the case of smaller values of h. Therefore, we have proved the first part of the statement.

Finally, we consider the second part of the statement. Let $\widehat{P}_{t,h}(\cdot|s,a)$ be the future model obtained by updating the current model $\widehat{P}(\cdot|s,a)$ with h *random* future samples (h samples drawn from $P(S|s,a)$ for each $(s,a) \in (S,A)$). Using a result given by Weissman et al. (2003, Theorem 2.1), we know that for all $s \in S$, with probability at

least $1-\delta$,

$$\max_{s,a}\|\widehat{P}_{t,h}(\cdot|s,a) - P(\cdot|s,a)\|_1 \leq \sqrt{\frac{2\ln(2^{|S|+1}|S||A|/\delta)}{n_{t,min}+h}},$$

where $n_{t,min} = \min_{s,a} n_t(s,a)$. Now, if we use the distance function,

$$d(\widehat{P}(\cdot|s,a), P(\cdot|s,a)) = \|\widehat{P}(\cdot|s,a) - P(\cdot|s,a)\|_1,$$

to define the h-reachable optimal function,

$$\begin{aligned}
|V^*(s_t) - V^{d*}_{\mathcal{L},t,h^*(\epsilon,\delta)}(s_t)| &\leq \frac{R_{max}}{1-\gamma}\max_{s,a}\|P^{d*}_{\mathcal{L},t,h}(\cdot|s,a) - P(\cdot|s,a)\|_1 \\
&= \frac{R_{max}}{1-\gamma}\max_{s,a}\min_{\widehat{P}\in\mathcal{M}_{\mathcal{L},t,h,(s,a)}}\|\widehat{P}(\cdot|s,a) - P(\cdot|s,a)\|_1 \\
&\leq \frac{R_{max}}{1-\gamma}\sqrt{\frac{2\ln(2^{|S|+1}|S||A|/\delta)}{n_{t,min}+h}},
\end{aligned}$$

The last inequality follows that the models reachable with h *random* samples $\widehat{P}_{t,h}(\cdot|s,a)$ are contained in a set of h-reachable models and the best h-reachable model $P^{d*}_{\mathcal{L},t,h}(\cdot|s,a)$ explicitly minimize the norm, resulting in that $P^{d*}_{\mathcal{L},t,h}(\cdot|s,a)$ is at least as good as $\widehat{P}_{t,h}(\cdot|s,a)$ in terms of the norm. The right-hand side of the above inequality becomes less than or equal to ϵ when $h \leftarrow h^*(\epsilon,\delta) = \frac{2R^2_{max}\ln(2^{|S|+1}|S||A|/\delta)}{\epsilon^2(1-\gamma)^2}$. Thus, we have the second part of the statement. □

Corollary 3.1. (Anytime error bound) With probability at least $1-\delta$, if $h \leq \frac{\ln(2^{|S|}|S||A|/\delta)}{\epsilon(1-\gamma)}$,

$$\epsilon_{t,h} = O\left(\sqrt[3]{\frac{|S||A|}{t(1-\gamma)^3}\ln\frac{|S||A|}{\delta}\ln\frac{2^{|S|}|S||A|}{\delta}}\right),$$

and otherwise,

$$\epsilon_{t,h} = O\left(\sqrt{\frac{h|S||A|}{t(1-\gamma)^2}\ln\frac{|S||A|}{\delta}}\right).$$

Proof. From Theorem 3.1, if $t = c\frac{z|S||A|}{\epsilon^2(1-\gamma)^2}\ln\frac{|S||A|}{\delta}$ with c being some fixed constant, $V^{\mathcal{A}}(s_t) \geq V^*_{\mathcal{L},t,h}(s_t) - \epsilon$ with probability at least $1-\delta$. Since this holds for all $t \geq 0$ with corresponding ϵ and δ, it implies that $\epsilon^2 \leq A\frac{z|S||A|}{t(1-\gamma)^2}\ln\frac{|S||A|}{\delta}$ with probability at least $1-\delta$. Substituting $z = \max(h, \frac{\ln(2^{|S|}|S||A|/\delta)}{\epsilon(1-\gamma)})$ yields the statement. □

The anytime T-step average loss is equal to $\frac{1}{T}\sum_{t=1}^{T}(1-\gamma^{T+1-t})\epsilon_{t,h,\delta}$. Since the errors considered in Theorem 3.1 and Corollary 3.3 are for an infinite horizon, the factor $(1-\gamma^{T+1-t})$ translates the infinite horizon error to the T-step finite horizon error (this can be seen when we modify the proof of Theorem 3.1 by replacing $\frac{1}{1-\gamma}$ with $\frac{1-\gamma^{T+1-t}}{1-\gamma}$).

Corollary 3.2. (Explicit exploration runtime) With probability at least $1-\delta$, the explicit exploration runtime of Algorithm 3.1 is

$$O(\frac{h|S||A|}{\epsilon(1-\gamma)\Pr[A_K]}\ln\frac{|S||A|}{\delta}) = O(\frac{h|S||A|}{\epsilon^2(1-\gamma)^2}\ln\frac{|S||A|}{\delta}),$$

where A_K is the escape event defined in the proof of Theorem 3.1.

Proof. The proof directly follows that of Theorem 3.1 with z. Compared to the sample complexity of Algorithm 3.1, z is replaced by h based on the proof of Theorem 3.1. □

B.5 Additional Experimental Example for Discrete Domain

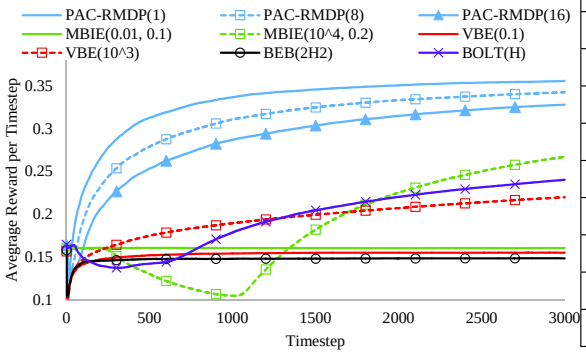
Figure B-1 shows the results in the main part of the thesis along with 10% and 90% values.Aside from the proposed algorithm, only BOLT gathered better rewards than a greedy algorithm while maintaining the claimed theoretical guarantee.

In this example, our proposed algorithm worked well and maintained its theoretical guarantee. One might consider the theoretical guarantee of PAC-RMDP, especially PAC-RMDP(1), to be too weak. Two things should be noted. First, the 1-reachable value function is not the value function that can be obtained with just one additional sample, but requires an additional sample for all $|S||A|$ state-action pairs. Second, in contrast to Bayesian optimality, the 1-reachable value function is not the value function *believed* to be obtained with $|S||A|$ additional samples, but is *possibly* reachable in terms of the unknown true world dynamics with the new samples.

However, it is certainly possible to devise a problem such that PAC-RMDP(1) is not guaranteed to conduct sufficient exploration. As an example, we consider a modified version of the five-state chain problem, where the probability of successfully moving away from the initial state is very small ($= 0.05$), thus requiring more extensive exploration. We modified the transition model as follows: Let a_1 be the optimal action that moves the agent away from the initial state. For $i = \{2, 3, 4, 5\}$, $\Pr(s_i, a_1, s_{\min(i+1,5)}) = 0.99$ and $\Pr(s_i, a_1, s_1) = 0.01$. For $i = 1$, $\Pr(s_i, a_1, s_{i+1)}) = 0.05$ and $\Pr(s_i, a_1, s_1) = 0.95$. For action a_2 and any s_i, $\Pr(s_i, a_2, s_1) = 1$. The numerical results for this example are shown in Figure B-2.2 As expected, the PAC-RMDP(1) algorithm often became stuck in the initial state.

B.6 Proofs of Theoretical Results for Algorithm 3.2

We assume that Algorithm 3.2 is used with the least square estimation, which determines $\mathcal{L}$. Because the true world dynamics are assumed to have the parametric form $P(s'|s, a) = \mathcal{N}(\theta^T \Phi(s, a), \sigma^2 I)$ with a known σ, their unknown aspect is attributed

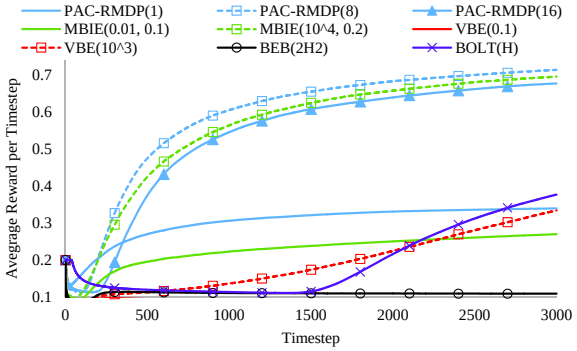


(a) Average of 1000 runs over all time steps

Algorithm	Average	10%	90%
PAC-RMDP(1)	0.357	0.332	0.378
PAC-RMDP(8)	0.343	0.321	0.365
PAC-RMDP(16)	0.328	0.305	0.321
MBIE(0.01, 0.1)	0.160	0.158	0.162
MBIE(20, 0.9)	0.160	0.158	0.162
MBIE(10^4, 0.2)	0.267	0.250	0.285
VBE(0.1)	0.155	0.152	0.158
VBE(0.99)	0.156	0.153	0.158
VBE(10^3)	0.220	0.207	0.232
BEB(2×148^2)	0.148	0.142	0.154
BOLT(148)	0.240	0.221	0.256

(b) Results for 1000 runs at time step 3000

Figure B-1: Average total reward per time step for the Chain Problem. The algorithm parameters are shown as PAC-RMDP(h), MBIE(ϵ, δ), VBE(δ), BEB(β), and BOLT(η).



(a) Average for 1000 runs over all time steps

Algorithm	Average	10%	90%
PAC-RMDP(1)	0.339	0.196	0.772
PAC-RMDP(8)	0.715	0.650	0.784
PAC-RMDP(16)	0.678	0.612	0.747
MBIE(0.01, 0.1)	0.270	0.260	0.279
MBIE(20, 0.9)	0.327	0.313	0.340
MBIE(10^4, 0.2)	0.697	0.634	0.752
VBE(0.1)	0.090	0.060	0.122
VBE(0.99)	0.094	0.061	0.126
VBE(10^3)	0.334	0.306	0.360
BEB(2×148^2)	0.108	0.103	0.113
BOLT(148)	0.377	0.314	0.441

(b) Results for 1000 runs at time step 3000

Figure B-2: Average total reward per time step for the modified Chain Problem. The algorithm parameters are shown as PAC-RMDP(h), MBIE(ϵ, δ), VBE(δ), BEB(β), and BOLT(η).

to the weight vector θ. Therefore, we discuss h-reachability in terms of $\hat{\theta}$ instead of $\widehat{P}$. For each i^{th} component, Let $\hat{\theta}^*_{(i),h,(s,a)}$ be the best h-reachable model parameter corresponding to the best h-reachable models, $\widehat{P}^*_{\mathcal{L},t,h}$ (we drop the index $\mathcal{L}, t$ and d for brevity); using the set $\hat{\theta}^*_{(i),h,(s,a)}$ for every (s,a) pair results in the h-reachable value function $V^{d*}_{\mathcal{L},t,h}$. Note that $\hat{\theta}_{(i)}$ is the current model parameter. In the following, we make a relatively strict assumption to simplify the analysis: when they are

not provided as inputs, the estimated values of L and $\Delta^{(i)}$ are correct in that they satisfy Assumption 3.2 and $\Delta^{(i)} \geq \sup_{s,a}|(\theta_{(i)} - \hat{\theta}_{(i)})^T \Phi_{(i)}(s,a)|$. This assumption can be relaxed by allowing the correctness to be violated with a constant probability. In such a case, we must force the random event to occur concurrently with the escape event, as discussed in the proof of Theorem 3.1 (the easiest way to do so is to take a union bound over the time steps until convergence). Furthermore, if we can specify the inputs L and $\Delta^{(i)}$, there is no need for this assumption.

Lemma 3.2. (Correctness of the h-reachable model interval) For the entire execution of Algorithm 3.2, for all state components $1 \leq i \leq n_s$, for all $t, h \geq 0$, and for all $(s,a) \in (S,A)$, the following inequality holds with probability at least $1 - \delta/2$:

$$\left|[\hat{\theta}_{(i)} - \hat{\theta}^*_{(i),h,(s,a)}]^T \Phi_{(i)}(s,a)\right| \leq I_h(\Phi_{(i)}(s,a), X_t).$$

Proof. Let $s_1^* \in S'_{(s,a)}$ be the future possible observation from which the current model parameter $\hat{\theta}_{(i)}$ is updated to $\hat{\theta}^*_{(i),1,(s,a)}$. Then,

$$\begin{aligned}\left|[\hat{\theta}^*_{(i),1,(s,a)} - \hat{\theta}_{(i)}]^T \Phi_{(i)}(s,a)\right| &= \left|\Phi^T_{(i)}(s,a)(X_t^T X_t)^{-1}\Phi_{(i)}(s,a)[s_1^* - \hat{\theta}^{*T}_{(i),1,(s,a)}\Phi_{(i)}(s,a)]\right| \\ &\leq \left|\Phi^T_{(i)}(s,a) D_t(\frac{1}{\lambda_{(1)}}, \ldots, \frac{1}{\lambda_{(n)}}) {U_t}^T \Phi_{(i)}(s,a)(\Delta^{(i)} + \varsigma(M)\sigma_{(i)})\right|.\end{aligned}$$

The first line follows directly from a result given by Cook (1977, Equation (5)). The second line is due to the following: with probability at least $1 - \frac{1}{2}e^{-\varsigma^2(M)/2}$,

$$\begin{aligned}s_1^* - \hat{\theta}^{*T}_{(i),1,(s,a)}\Phi_{(i)}(s,a) &\leq \theta^T_{(i)}\Phi_{(i)}(s,a) - \hat{\theta}^{*T}_{(i),1,(s,a)}\Phi_{(i)} + \varsigma(M)\sigma_{(i)} \\ &\leq |\theta^T_{(i)}\Phi_{(i)}(s,a) - \hat{\theta}^{*T}_{(i),1,(s,a)}\Phi_{(i)}(s,a)| + \varsigma(M)\sigma_{(i)} \\ &\leq |\theta^T_{(i)}\Phi_{(i)}(s,a) - \hat{\theta}^T_{(i)}\Phi_{(i)}(s,a)| + \varsigma(M)\sigma_{(i)} \\ &\leq \Delta^{(i)} + \varsigma(M)\sigma_{(i)}\end{aligned}$$

where the first inequality follows that $\Pr(s_{t+1} > \theta_{(i)}^T \Phi_{(i)}(s,a) + \varsigma(M)\sigma_{(i)}) < \frac{1}{2}e^{-\varsigma^2(M)/2}$ and the third inequality follows the choice of the distance function d (i.e., the mean prediction with the best h reachable model is at least as good as that of the best $h-1$ model). We then separate the above into two terms with large and small eigenvalues: with probability at least $1 - \frac{1}{2}e^{-\varsigma^2(M)/2}$,

$$\begin{aligned}\left|[\hat{\theta}^*_{(i),1(s,a)} - \hat{\theta}_{(i)}]^T \Phi_{(i)}(s,a)\right| \leq & |\Phi^T_{(i)}(s,a) U_t D_t (\frac{1}{\lambda_{(1)}}, \ldots, \frac{1}{\lambda_{(j)}}, 0, \ldots, 0) {U_t}^T \Phi_{(i)}(s,a)(\Delta^{(i)} + \varsigma(M)\sigma_{(i)}) \\ & + \Phi^T_{(i)}(s,a) U_t D_t (0, \ldots, 0, \frac{1}{\lambda_{(j+1)}}, \ldots, \frac{1}{\lambda_{(n)}}) {U_t}^T \Phi_{(i)}(s,a)(\Delta^{(i)} + \varsigma(M)\sigma_{(i)})|.\end{aligned}$$

With w_t, we can rewrite part of the second term as $UD(0, \ldots, 0, \frac{1}{\lambda_{(j+1)}}, \ldots, \frac{1}{\lambda_{(n)}})U^T = UD(\frac{1}{\lambda_{(1)}}, \ldots, \frac{1}{\lambda_{(n)}})U^T w_t$. Then, with g_t and z_t, with probability at least $1 - \frac{1}{2}e^{-\varsigma^2(M)/2}$,

$$\left|[\hat{\theta}^*_{(i),1,(s,a)} - \hat{\theta}_{(i)}]^T \Phi_{(i)}(s,a)\right| \leq (\Delta^{(i)} + \varsigma(M)\sigma_{(i)}) \left|\Phi^T_{(i)}(s,a) g_t \Phi_{(i)}(s,a) + \Phi^T_{(i)}(s,a) z_t w_t \Phi_{(i)}(s,a)\right|.$$

Thus, by applying the union bound for h, with probability at least $1 - \frac{h}{2}e^{-\varsigma^2(M)/2}$,

$$\begin{aligned}\left|[\hat{\theta}^*_{(i),h,(s,a)} - \hat{\theta}_{(i)}]^T \Phi_{(i)}(s,a)\right| &\leq h \left|[\hat{\theta}^*_{(i),1,(s,a)} - \hat{\theta}_{(i)}]^T \Phi_{(i)}(s,a)\right| \\ &\leq h(\Delta^{(i)} + \varsigma(M)\sigma_{(i)}) \left|\Phi^T_{(i)}(s,a) g_t \Phi_{(i)}(s,a) + \Phi^T_{(i)}(s,a) z_t w_t \Phi_{(i)}(s,a)\right| \\ &\leq I_h(\Phi_{(i)}(s,a), X_t).\end{aligned}$$

For n_s components, the above inequality holds with probability at least $1 - \frac{n_s h}{2}e^{-\varsigma^2(M)/2}$ (union bound). For all $M \geq 1$, the above inequality holds with probability at least $1 - \frac{n_s h}{2}\sum_{M=1}^{\infty} e^{-\varsigma^2(M)/2}$ (union bound). Substituting $\varsigma(M) = \sqrt{2\ln(\pi^2 M^2 n_s h/(6\delta))}$, we obtain the statement. □

In Lemma 3.3 and Theorem 3.2, following previous work [44, 27], we assume that an exact planning algorithm is accessible. This assumption will be relaxed by using a planning method that provides an error bound. We also assume that $R_{max} \leq c_1$, $\Delta^{(i)} \leq c_2$, and $\|\theta\| \leq c_3$ for some fixed constants c_1, c_2, and c_3. Removing this assumption results in these quantities appearing in the sample complexity, but produces no exponential dependence (thus, the sample complexity remains polynomial). We assume that $M = O(\text{the number of samples})$, meaning that a planing algorithm calls $\mathbf{I}_h$ every iteration at most for a constant number of times. In the following, we use

$\bar{n}$ to represent the average value of $\{n_{(1)}, ..., n_{(n_S)}\}$. Before analyzing the proposed algorithm, we re-derive the sample complexity of an existing PAC-MDP algorithm [44, 27] for our problem setting.

Lemma 3.3. (Sample complexity of PAC-MDP) With an appropriate parameter setting, the PAC-MDP algorithm proposed by Strehl and Littman [44] and Li et al. [27] has the following sample complexity:

$$\tilde{O}\left(\frac{n_S^2 \bar{n}^2}{\epsilon^5 (1-\gamma)^{10}}\right).$$

Proof. The proof follows directly from Theorems 3.1 and 3.3 in the previous work of Li et al. [27]. The only difference is that we need to take a union bound of different components $\Phi_{(i)}$ with varying domains, codomains and dimensions $n_{(s)}$. □

Theorem 3.2. (PAC-RMDP) Let $\mathcal{A}_t$ be a policy of Algorithm 3.2. Let

$$z = \max(h^2 \ln \frac{m^2 n_s h}{\delta}, \frac{L^2 n_S \bar{n} \ln^2 m}{\epsilon^3} \ln \frac{n_S}{\delta}).$$

Then, for all $\epsilon > 0$, for all $\delta = (0, 1)$, and for all $h \geq 0$,

1) for all but at most $m' = O\left(\frac{zL^2 n_S \bar{n} \ln^2 m}{\epsilon^3 (1-\gamma)^2} \ln^2 \frac{n_S}{\delta}\right)$ time steps (with $m \leq m'$),

$$V^{\mathcal{A}_t}(s_t) \geq V^*_{\mathcal{L},t,h}(s_t) - \epsilon,$$

with probability at least $1 - \delta$, *and*

2) there exists $h^*(\epsilon, \delta) = O(\mathcal{P}(1/\epsilon, 1/\delta, 1/(1-\gamma), |\text{MDP}|))$ such that

$$|V^*(s_t) - V^*_{\mathcal{L},t,h^*(\epsilon,\delta)}(s_t)| \leq \epsilon$$

with probability at least $1-\delta$.

Proof. Let $\tilde{V}^A$ be the internal value function used in Algorithm 3.2. We prove the statement by following the structure of the proof of Theorem 3.1. Define K, m, A_K, $\overline{V}$, and H in the same manner as in the proof of Theorem 3.1, and let the vector consisting of n_S estimation error intervals be $\mathbf{ER}(s,a) = (|(\theta_{(1)} - \hat{\theta}_{(1)})^T \Phi_{(1)}(s,a)|, ..., |(\theta_{(n_s)} - \hat{\theta}_{(n_s)})^T \Phi_{(n_s)}(s,a)|$. By following the proof of Theorem 3.1, with probability at least $1-\delta/2$ (due to Lemma 3.2),

$$
\begin{aligned}
V^{\mathcal{A}}(s_t) &\geq \tilde{V}^{\mathcal{A}}(s_t) - \frac{R_{max}}{1-\gamma} Pr(A_k) - \frac{\epsilon}{3} - L\left(\max_{s,a}\|\mathbf{I}_h(s,a,X_{m'})\| + \max_{s,a}\|\mathbf{ER}(s,a)\|\right) \\
&\geq V^*_{\mathcal{L},t,h}(s_t) - \frac{c_1}{1-\gamma} Pr(A_k) - \frac{\epsilon}{3} - L\left(\max_{s,a}\|\mathbf{I}_h(s,a,X_{m'})\| + \max_{s,a}\|\mathbf{ER}(s,a)\|\right).
\end{aligned}
$$

In the second line, we used the assumption $R_{max} \leq c_1$. In the first line, $\max_{s,a} L\|\mathbf{I}_h(s, a, X_t)\|$ is the difference between $\tilde{V}^{\mathcal{A}}(s_t)$ and $V^*_{\mathcal{L},t,0}(s_t)$, and $\max_{s,a} L\|\mathbf{ER}(s,a)\|$ is the difference between $V^*_{\mathcal{L},t,0}(s_t)$ and $V^{\mathcal{A}}$. The second line follows from the fact that $\tilde{V}^{\mathcal{A}} \geq V^*_{\mathcal{L},t,h}(s_t)$ because of the correctness of I_h shown in Lemma 3.2 and the assignment of the most optimistic value within the interval $\mathbf{I}_h$ (based on Assumptions 3.1 and 3.2). We now impose an upper bound on $\|\mathbf{I}_h(s,a,X_t)\|$ and $\|\mathbf{ER}(s,a)\|$. Following a proof given by Li et al. (2011, Theorem 1) with the assumption $\Delta^{(i)} \leq c_2$ and $\|\theta\| \leq c_3$, with probability at least $1 - \frac{\delta}{4n_S}$,

$$
\begin{aligned}
&|(\theta_{(i)} - \hat{\theta}_{(i)})^T \Phi_{(i)}(s,a)| \leq \|\bar{q}\| \Delta_E(\hat{\theta}) + \|\bar{u}\| \\
&\leq \frac{2c_3\sqrt{n_{(i)} \ln m}}{m^{1/4}} (24 c_2 \ln \frac{8n_S}{\delta})^{1/4} + \frac{(2c_3\sqrt{\ln m} + 5)\sqrt{n_{(i)}}}{\sqrt{m}} \\
&\leq O\left(\frac{(n_{(i)} \ln m)^{1/2} (\ln(n_S/\delta))^{1/4}}{m^{1/4}}\right),
\end{aligned}
$$

where $\|\bar{q}\|, \|\bar{u}\|$ and $\Delta_E(\hat{\theta})$ are as defined by Li et al. [27]. Since $\Phi^T_{(i)} z_t (s_{t+1} - \hat{\theta}^T_{t+1} \Phi_{(i)}) = \hat{\theta}_{t+1} - \hat{\theta}_t$, there exist $\hat{\theta}$ and $\hat{\theta}'$ such that $\left\| \Phi^T_{(i)}(s,a) z_t (\Delta^{(i)} + \varsigma(M)\sigma_{(i)}) \right\| \leq \|\hat{\theta} - \hat{\theta}'\| \leq \|\hat{\theta}\| + \|\hat{\theta}'\| \leq 2c_3$, where we use the assumption $\|\theta\| \leq c_3$. Then, following the

proofs of Lemmas 11, 12, and 13 given by Auer [4],

$$\begin{aligned}\frac{I_h(\Phi_{(i)}(s,a),X_t)}{h} &\leq (\Delta^{(i)}+\varsigma(M)\sigma_{(i)})\sum_{j:\lambda_j\geq 1}\frac{\Phi_j^2}{\lambda_j}+\|\hat{\theta}-\hat{\theta}'\|\sqrt{\sum_{j:\lambda_j<1}\Phi_j^2}\\ &\leq \frac{20(c_2+\sqrt{2\ln(\pi^2M^2n_sh/(6\delta))}\sigma_{(i)})n\ln(m)}{m}+2c_3\sqrt{\frac{20n_{(i)}}{m}}\\ &\leq O\left(\frac{\sqrt{n_{(i)}}}{\sqrt{m}}\ln m\sqrt{\ln(m^2n_sh/(6\delta))}\right).\end{aligned}$$

If $h \leq O(\frac{m^{1/2}(\ln n_S/\delta)^{1/4}}{(\ln m)^{1/2}(\ln(m^2n_sh/(6\delta)))^{1/2}})$, with probability at least $1-n_s\frac{\delta}{4n_s}-\delta/2$,

$$V^{\mathcal{A}}(s_t) \geq V^*_{\mathcal{L},t,h}(s_t)-\frac{c_1\Pr(A_k)}{1-\gamma}-\frac{\epsilon}{3}-O\left(\frac{Ln_S^{1/2}\bar{n}^{1/2}(\ln m)^{1/2}(\ln(n_S/\delta))^{1/4}}{m^{1/4}}\right).$$

If $h > O(\frac{m^{1/2}(\ln n_S/\delta)^{1/4}}{(\ln m)^{1/2}(\ln(m^2n_sh/(6\delta)))^{1/2}})$, with probability at least $1-n_s\frac{\delta}{4n_s}-\delta/2$,

$$V^{\mathcal{A}}(s_t) \geq V^*_{\mathcal{L},t,h}(s_t)-\frac{c_1\Pr(A_k)}{1-\gamma}-\frac{\epsilon}{3}-O\left(\frac{Lhn_S^{1/2}\bar{n}^{1/2}}{\sqrt{m}}\ln m\sqrt{\ln(m^2n_sh/(6\delta))}\right).$$

To have $\epsilon/3$ in the last term, we fix $m = O(\frac{L^4n_S^2\bar{n}^2\ln^4 m}{\epsilon^4}\ln\frac{n_S}{\delta})$ for the former case, and $m = O(\frac{L^2h^2n_S\bar{n}\ln^2 m\ln(m^2n_sh/(6\delta))}{\epsilon^2})$ for the latter case. Then, the rest of the first part of the statement follows from the proof of Theorem 1. That is, we can show that by applying the Chernoff bound, the escape event happens no more than the sample complexity in the statement with probability $1-\delta/2$ unless the term $\frac{c_1\Pr(A_k)}{1-\gamma}$ is negligible. Taking union bound on the failure probability, we obtain the sample complexity in the statement with probability at leat $1-\delta$.

Finally, we consider the second part of the statement, following the proof in Theorem 3.1. Let $\hat{\theta}_{(i),h,(s,a)}$ be the future model parameter obtained by updating the current model $\hat{\theta}_{(i)}$ with h *random* future samples (h samples drawn from $P(S|s,a)$ for each $(s,a) \in (S,A)$). Based on the first part of the proof, $|(\theta_{(i)}-\hat{\theta}_{(i),h,(s,a)})\hat{}\mathrm{T}\ \Phi_{(i)}(s,a)| \leq O\left(\frac{(n_{(i)}\ln(n_{min}+h))^{1/2}(\ln(n_S/\delta))^{1/4}}{(n_{min}+h)^{1/4}}\right)$ with probability at least $1-\delta$. Since $|(\theta_{(i)}-\hat{\theta}^*_{(i),h,(s,a)})^T\Phi_{(i)}(s,a)| \leq |(\theta_{(i)}-\hat{\theta}_{(i),h,(s,a)})^T\Phi_{(i)}(s,a)|$ (this directly follows the definition of $\hat{\theta}^*_{(i),h,(s,a)}$ and the choice of the distance function d), this implies that

$h^*(\epsilon, \delta) = O(\frac{L^4 n_S^2 \bar{n}^2 \ln^2 m}{\epsilon^4} \ln \frac{n_S}{\delta})$ is sufficient. □

Corollary 3.3. (Anytime error bound) With probability at least $1-\delta$, if $h^2 \ln \frac{m^2 n_s h}{\delta} \leq \frac{L^2 n_S \bar{n} \ln^2 m}{\epsilon^3} \ln \frac{n_S}{\delta}$,

$$\epsilon_{t,h} = O\left(\sqrt[5]{\frac{L^4 n_S^2 \bar{n}^2 \ln^2 m}{t(1-\gamma)} \ln^3 \frac{n_S}{\delta}}\right),$$

and otherwise,

$$\epsilon_{t,h} = O\left(\frac{h^2 L^2 n_S \bar{n} \ln^2 m}{t(1-\gamma)} \ln^2 \frac{n_S}{\delta}\right).$$

Proof. The proof follows directly from Theorem 3.2 and the proof of Corollary 3.1. □

As in the discrete case, the anytime T-step average loss can be computed by summing the anytime errors as $\frac{1}{T}\sum_{t=1}^{T}(1-\gamma^{T+1-t})\epsilon_{t,h,\delta}$. In addition, we can derive the explicit exploration runtime.

Corollary 3.6. (Explicit exploration runtime) With probability at least $1-\delta$, the explicit exploration runtime of Algorithm 3.2 is

$$O\left(\frac{h^2 L^2 n_S \bar{n} \ln m}{\epsilon^2 \Pr[A_k]} \ln^2 \frac{n_S}{\delta} \ln \frac{m^2 n_s h}{\delta}\right) = O\left(\frac{h^2 L^2 n_S \bar{n} \ln m}{\epsilon^3 (1-\gamma)} \ln^2 \frac{n_S}{\delta} \ln \frac{m^2 n_s h}{\delta}\right),$$

where A_K is the escape event defined in the proof of Theorem 3.2.

Proof. The proof follows that of Theorem 3.2. Compared to the sample complexity of Algorithm 3.2, z is replaced by h based on the proof of Theorem 3.2. □

B.7 Experimental Settings for Continuous Domain

For each problem used in the main part of the thesis, we present more detailed descriptions of the experimental settings.

Mountain Car

In the mountain car problem, the reward is negative everywhere except at the goal. To reach the goal, the agent must first travel far away, and must explore the world to learn this mechanism. To require a greater degree of exploration, we modify the original problem as follows: The agent obtains a reward equal to -0.9 around the initial position (position = [-0.6, 0.4]), and -1.0 everywhere else but at the goal. At the start of each episode, the agent is always at the bottom of the valley (position = -0.5) with zero velocity. Moreover, a small amount of Gaussian noise with standard deviation 0.001 is added to the velocity. Our model uses 10 grids of residual basis functions over the control signal and velocity as features. For the planning phase, we use a fitted value iteration with a 30×30 grid of residual basis functions. We set $\Delta^{(i)}$ and the corresponding parameter in the PAC-MDP algorithm to be 0.14, because the velocity is bounded in $[-0.07, 0.07]$. Each episode consists of 2000 steps, and we conduct simulations for 100 episodes.

Simulated HIV treatment

This problem is described by a set of six ordinary differential equations [12]. An action corresponds to whether the agent administers two treatments (RTIs and PIs) to patients (thus, there are four actions). Two types of exploration are required: one to learn the effect of using treatments on viruses, and another to learn the effect of not using treatments on immune systems. Learning the former is necessary to reduce the population of viruses, but the latter is required to prevent the overuse of treatments, which weakens the immune system. We select the initial state to be unhealthy, following Ernst et al. [12] and Pazis and Parr [33]. As in previous work, we assume that *noise-free* data can be obtained every five days. Unlike past studies,

we assume that *noisy* data can be obtained a day after each instance of noise-free data is collected, with the noise term being $\zeta' \sim \mathcal{N}(0, 0.1)$. We add another noise term to represent the model error with $\zeta \sim \mathcal{N}(0, 0.01)$ for each dynamic state. For the model, we use the six states and the multiple of any two of these six states as features (i.e., the number of features is $6 + \binom{6}{2}$). For planning, we use a greedy roll-out method, as described by Adams et al. (2004, Section 5). We set $\Delta^{(i)}$ and the corresponding parameter in the PAC-MDP algorithm to be the average error among all the predictions and observations. Each episode consists of 1000 days, and we conduct simulations for 30 episodes.

Bibliography

[1] Pieter Abbeel and Andrew Y Ng. Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning (ICML)*, 2005.

[2] BM Adams, HT Banks, Hee-Dae Kwon, and Hien T Tran. Dynamic multidrug therapies for HIV: Optimal and STI control approaches. *Mathematical Biosciences and Engineering*, 1(2):223–241, 2004.

[3] Mauricio Araya-López, Vincent Thomas, and Olivier Buffet. Near-optimal BRL using optimistic local transitions. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[4] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research (JMLR)*, 3:397–422, 2002.

[5] Andrey Bernstein and Nahum Shimkin. Adaptive-resolution reinforcement learning with polynomial exploration in deterministic domains. *Machine learning*, 81 (3):359–397, 2010.

[6] Emma Brunskill. Bayes-optimal reinforcement learning for discrete uncertainty domains. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.

[7] S. Bubeck, G. Stoltz, and J. Y. Yu. Lipschitz bandits without the Lipschitz constant. In *Algorithmic Learning Theory*, pages 144–158. Springer, 2011.

[8] R. G. Carter, J. M. Gablonsky, A. Patrick, C. T. Kelley, and O. J. Eslinger. Algorithms for noisy problems in gas transmission pipeline optimization. *Optimization and engineering*, 2(2):139–157, 2001.

[9] R Dennis Cook. Detection of influential observation in linear regression. *Technometrics*, pages 15–18, 1977.

[10] N. De Freitas, A. J. Smola, and M. Zoghi. Exponential regret bounds for Gaussian process bandits with deterministic observations. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[11] L. C. W. Dixon. *Global optima without convexity*. Numerical Optimisation Centre, Hatfield Polytechnic, 1977.

[12] Damien Ernst, Guy-Bart Stan, Jorge Goncalves, and Louis Wehenkel. Clinical data based optimal STI strategies for HIV: a reinforcement learning approach. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006.

[13] Claude-Nicolas Fiechter. Efficient reinforcement learning. In *Proceedings of the seventh annual ACM conference on Computational learning theory (COLT)*, 1994.

[14] J. Gardner, M. Kusner, K. Weinberger, and J. Cunningham. Bayesian Optimization with Inequality Constraints. In *Proceedings of The 31st International Conference on Machine Learning (ICML)*, pages 937–945, 2014.

[15] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *The Journal of Machine Learning Research (JMLR)*, 11:1563–1600, 2010.

[16] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[17] Kirthevasan Kandasamy, Jeff Schneider, and Barnabas Poczos. High dimensional Bayesian optimisation and bandits via additive models. *arXiv preprint arXiv:1503.01673*, 2015.

[18] Kenji Kawaguchi. Bounded optimal exploration in MDP. In *In Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[19] Kenji Kawaguchi and Mauricio Araya. A greedy approximation of Bayesian reinforcement learning with probably optimistic transition model. In *Proceedings of AAMAS 2013 workshop on adaptive learning agents*, pages 53–60, 2013.

[20] Kenji Kawaguchi, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Bayesian optimization with exponential convergence. In *In Advances in Neural Information Processing (NIPS)*, 2015. To Appear.

[21] Michael Kearns and Satinder Singh. Finite-sample convergence rates for Q-learning and indirect algorithms. In *Proceedings of Advances in neural information processing systems (NIPS)*, 1999.

[22] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

[23] J Zico Kolter and Andrew Y Ng. Near-Bayesian exploration in polynomial time. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, 2009.

[24] D. E. Kvasov, C. Pizzuti, and Y. D. Sergeyev. Local tuning and partition strategies for diagonal GO methods. *Numerische Mathematik*, 94(1):93–106, 2003.

[25] Lihong Li. *A unifying framework for computational reinforcement learning theory*. PhD thesis, Rutgers, The State University of New Jersey, 2009.

[26] Lihong Li. Sample complexity bounds of exploration. In *Reinforcement Learning*, pages 175–204. Springer, 2012.

[27] Lihong Li, Michael L Littman, Thomas J Walsh, and Alexander L Strehl. Knows what it knows: a framework for self-aware learning. *Machine learning*, 82(3):399–443, 2011.

[28] D. Q. Mayne and E. Polak. Outer approximation algorithm for nondifferentiable optimization problems. *Journal of Optimization Theory and Applications*, 42(1): 19–30, 1984.

[29] D. B. McDonald, W. J. Grantham, W. L. Tabor, and M. J. Murphy. Global and local optimization using radial basis function response surface models. *Applied Mathematical Modelling*, 31(10):2095–2110, 2007.

[30] R. H. Mladineo. An algorithm for finding the global maximum of a multimodal, multivariate function. *Mathematical Programming*, 34(2):188–200, 1986.

[31] R. Munos. Optimistic optimization of deterministic functions without the knowledge of its smoothness. In *Proceedings of Advances in neural information processing systems (NIPS)*, 2011.

[32] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, page 521, 2012.

[33] Jason Pazis and Ronald Parr. PAC Optimal Exploration in Continuous Space Markov Decision Processes. In *Proceedings of the 27th AAAI conference on Artificial Intelligence (AAAI)*, 2013.

[34] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2004.

[35] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[36] Stuart J Russell and Devika Subramanian. Provably bounded-optimal agents. *Journal of Artificial Intelligence Research (JAIR)*, pages 575–609, 1995.

[37] B. O. Shubert. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388, 1972.

[38] Herbert A Simon. *Models of bounded rationality, volumes 1 and 2*. MIT press, 1982.

[39] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 2951–2959, 2012.

[40] Jonathan Sorg, Satinder Singh, and Richard L Lewis. Variance-based rewards for approximate Bayesian reinforcement learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

[41] N. Srinivas, A. Krause, M. Seeger, and S. M. Kakade. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 1015–1022, 2010.

[42] Alexander L Strehl. *Probably approximately correct (PAC) exploration in reinforcement learning*. PhD thesis, Rutgers University, 2007.

[43] Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

[44] Alexander L Strehl and Michael L Littman. Online linear regression and its application to model-based reinforcement learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1417–1424, 2008.

[45] Alexander L Strehl, Lihong Li, and Michael L Littman. Incremental model-based learners with formal learning-time guarantees. In *Proceedings of the 22th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.

[46] Malcolm Strens. A Bayesian framework for reinforcement learning. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, 2000.

[47] R. G. Strongin. Convergence of an algorithm for finding a global extremum. *Engineering Cybernetics*, 11(4):549–555, 1973.

[48] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved November 30, 2014, from `http://www.sfu.ca/~ssurjano`, 2014.

[49] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[50] Thomas J Walsh, Sergiu Goschin, and Michael L Littman. Integrating Sample-Based Planning and Model-Based Reinforcement Learning. In *Proceedings of the 24th AAAI conference on Artificial Intelligence (AAAI)*, 2010.

[51] Z. Wang, B. Shakibi, L. Jin, and N. de Freitas. Bayesian Multi-Scale Optimistic Optimization. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTAT)*, pages 1005–1014, 2014.

[52] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, and Nando De Freitas. Bayesian optimization in high dimensions via random embeddings. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1778–1784. AAAI Press, 2013.

[53] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the L1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.

[54] Shlomo Zilberstein. Metareasoning and bounded rationality. In *Proceedings of the AAAI workshop on Metareasoning: Thinking about Thinking*, 2008.

[55] J. W. Zwolak, J. J. Tyson, and L. T. Watson. Globally optimised parameters for a model of mitotic control in frog egg extracts. *IEEE Proceedings-Systems Biology*, 152(2):81–92, 2005.